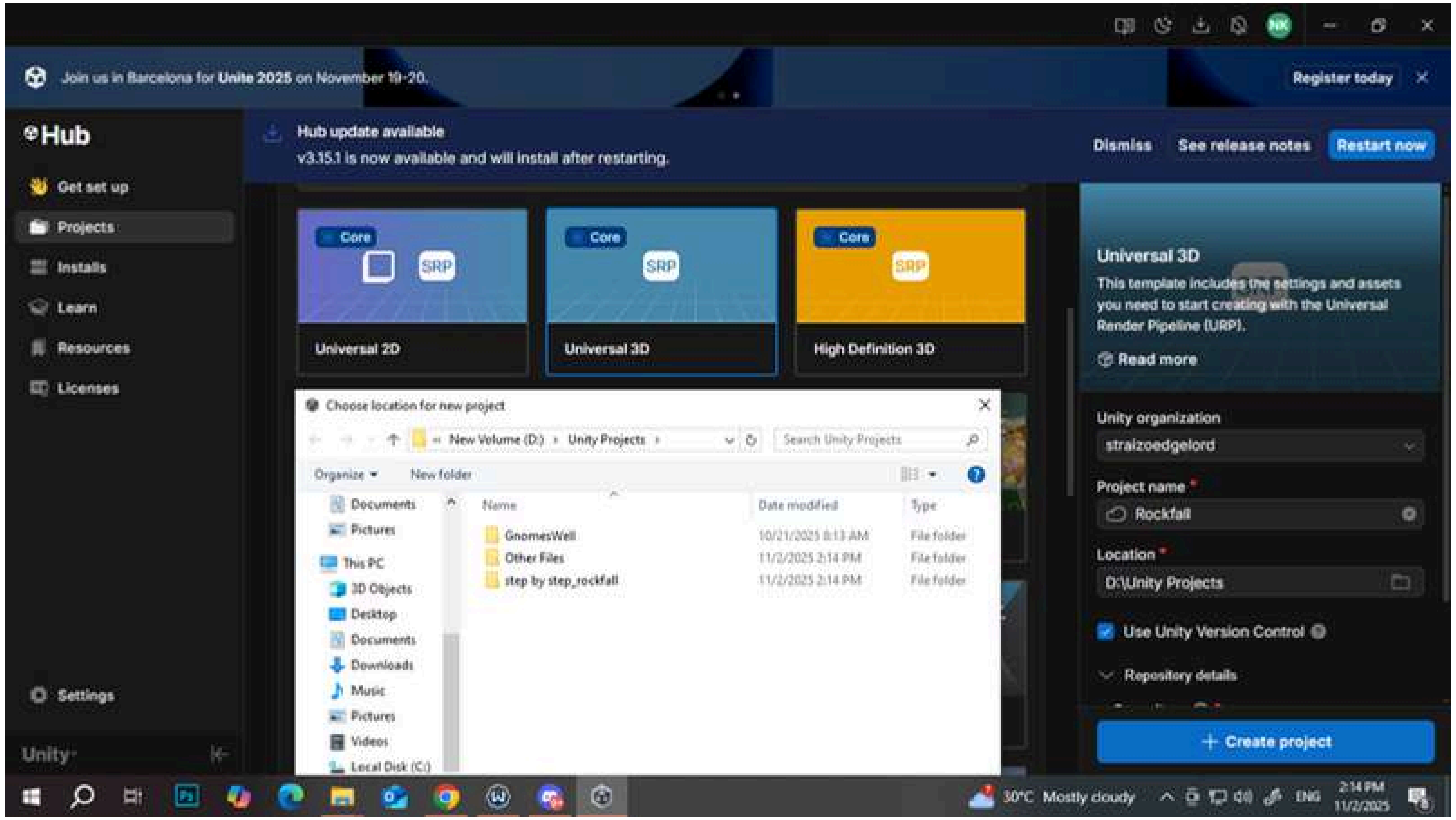
ROCKFALL

CHUMILANG.RAQUEPO.SEBASTIAN

START

STEP 1

Create game in 3D through Unity Hub



Join us in Barcelona for **Unite 2025** on November 19-20.

Hub update available
v3.15.1 is now available and will install after restarting.

Universal 3D
This template includes the settings and assets you need to start creating with the Universal Render Pipeline (URP).

Unity organization: straizoedgelord
Project name *: Rockfall
Location *: D:\Unity Projects
[✓] Use Unity Version Control

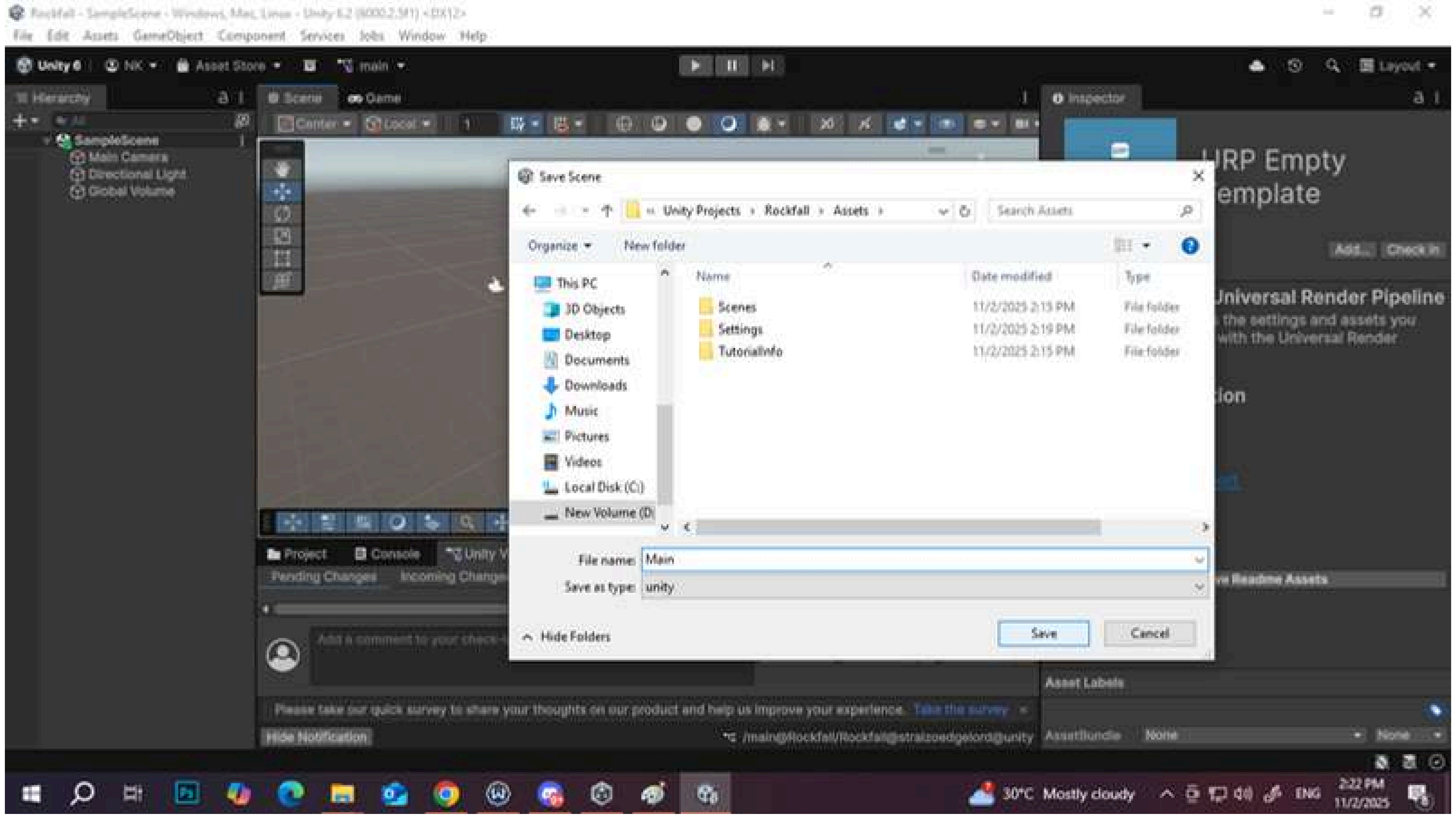
Choose location for new project

New Volume (D:) › Unity Projects ›

Name	Date modified	Type
GnomesWell	10/21/2025 8:13 AM	File folder
Other Files	11/2/2025 2:14 PM	File folder
step by step_rockfall	11/2/2025 2:14 PM	File folder

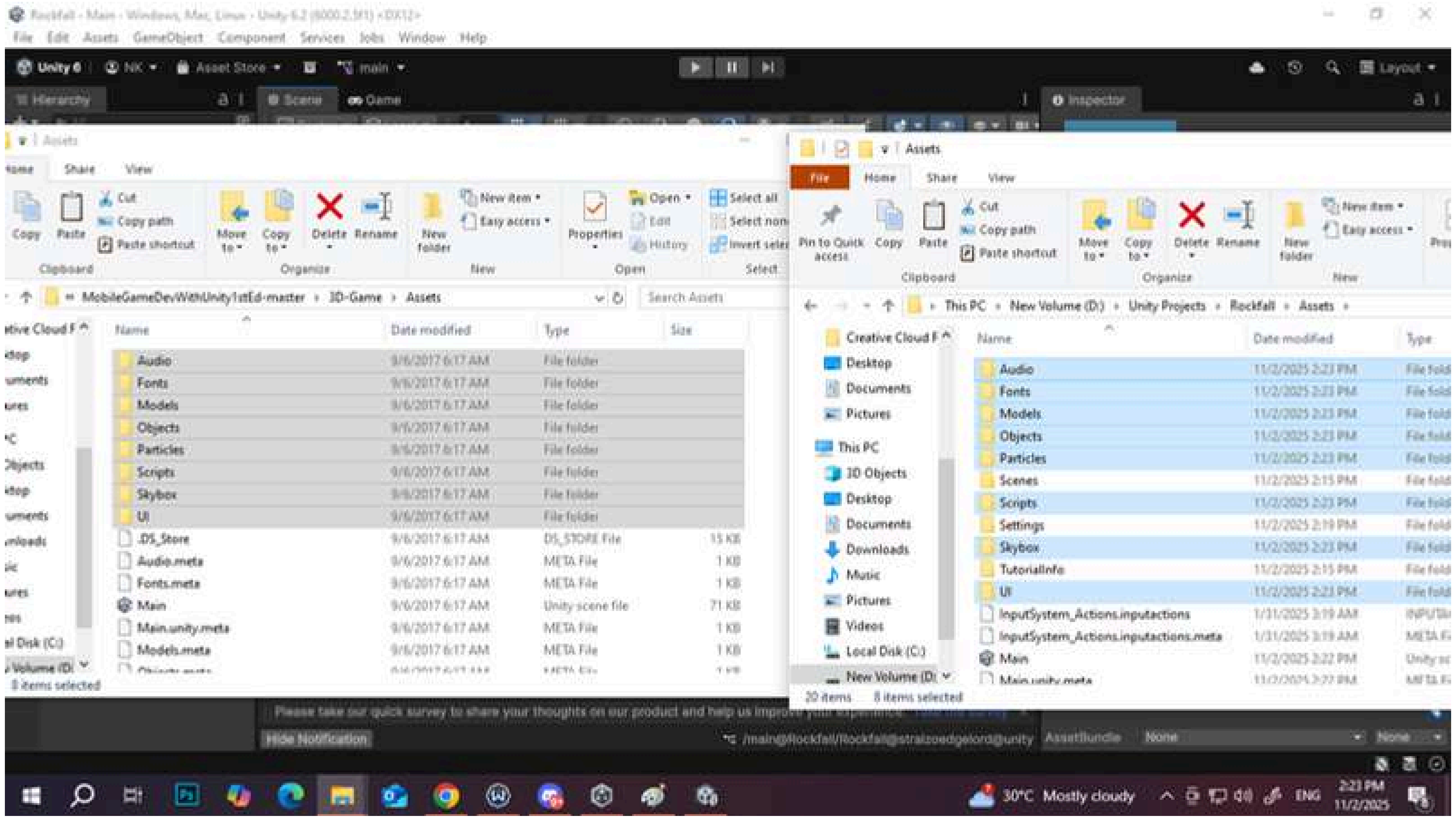
STEP 2

Save scene as Main



STEP 3

Import assets from



MobileGameDevWithUnity1stEd-master › 3D-Game › Assets

Name	Date modified	Type	Size
Audio	9/6/2017 6:17 AM	File folder	
Fonts	9/6/2017 6:17 AM	File folder	
Models	9/6/2017 6:17 AM	File folder	
Objects	9/6/2017 6:17 AM	File folder	
Particles	9/6/2017 6:17 AM	File folder	
Scripts	9/6/2017 6:17 AM	File folder	
Skybox	9/6/2017 6:17 AM	File folder	
UI	9/6/2017 6:17 AM	File folder	
.DS_Store	9/6/2017 6:17 AM	DS_STORE File	15 KB
Audio.meta	9/6/2017 6:17 AM	META File	1 KB
Fonts.meta	9/6/2017 6:17 AM	META File	1 KB
Main	9/6/2017 6:17 AM	Unity scene file	71 KB
Main.unity.meta	9/6/2017 6:17 AM	META File	1 KB
Models.meta	9/6/2017 6:17 AM	META File	1 KB

8 items selected

This PC › New Volume (D:) › Unity Projects › Rockfall › Assets

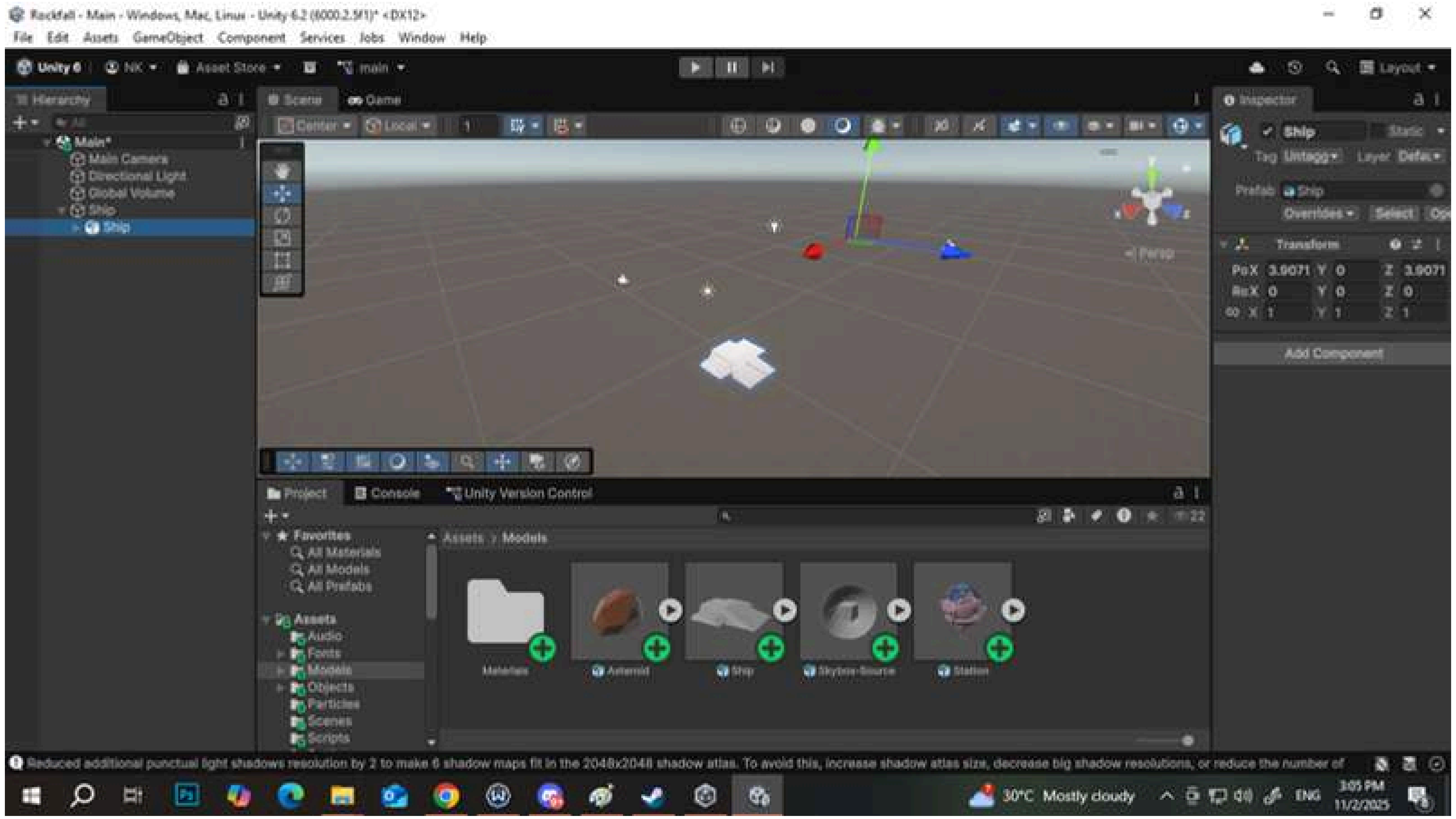
Name	Date modified	Type
Audio	11/2/2025 2:23 PM	File fold
Fonts	11/2/2025 2:23 PM	File fold
Models	11/2/2025 2:23 PM	File fold
Objects	11/2/2025 2:23 PM	File fold
Particles	11/2/2025 2:23 PM	File fold
Scenes	11/2/2025 2:15 PM	File fold
Scripts	11/2/2025 2:23 PM	File fold
Settings	11/2/2025 2:19 PM	File fold
Skybox	11/2/2025 2:23 PM	File fold
TutorialInfo	11/2/2025 2:15 PM	File fold
UI	11/2/2025 2:23 PM	File fold
InputSystem_Actions.inputactions	1/31/2025 3:19 AM	INPUTA
InputSystem_Actions.inputactions.meta	1/31/2025 3:19 AM	META F
Main	11/2/2025 2:22 PM	Unity sc

20 items 8 items selected

Please take our quick survey to share your thoughts on our product and help us improve your experience.

STEP 4

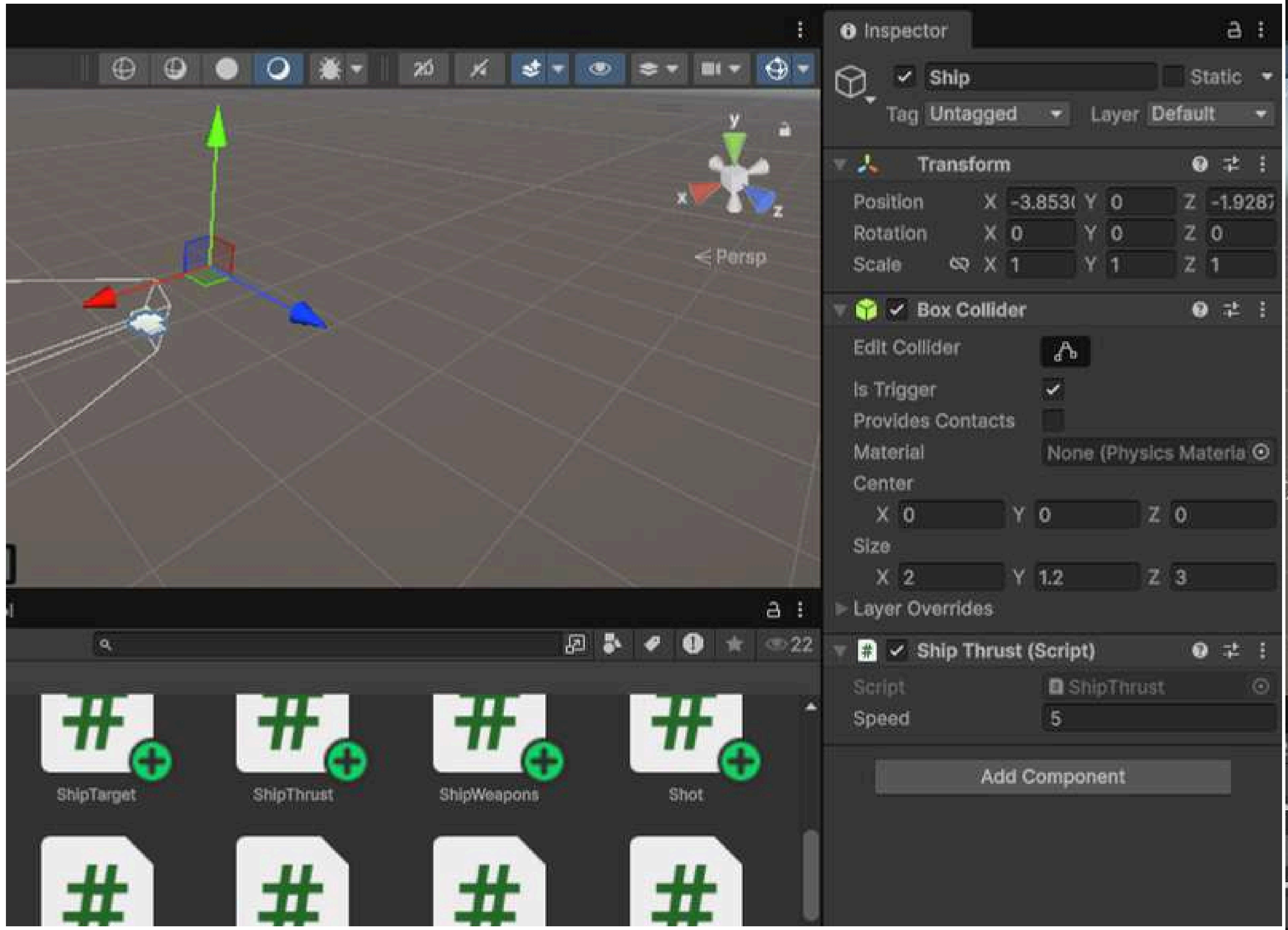
1. Make new GameObject, rename as Ship and drag imported ship model to it as child
2. Rename ship model as Graphics then reset position



Reduced additional punctual light shadows resolution by 2 to make 6 shadow maps fit in the 2048x2048 shadow atlas. To avoid this, increase shadow atlas size, decrease big shadow resolutions, or reduce the number of

STEP 5

1. Add box collider, set Is Trigger on and size to 2, 1.2, 3 then add ship thrust script
2. Add smooth follow script to main camera to follow the ship, set target as Ship under the script



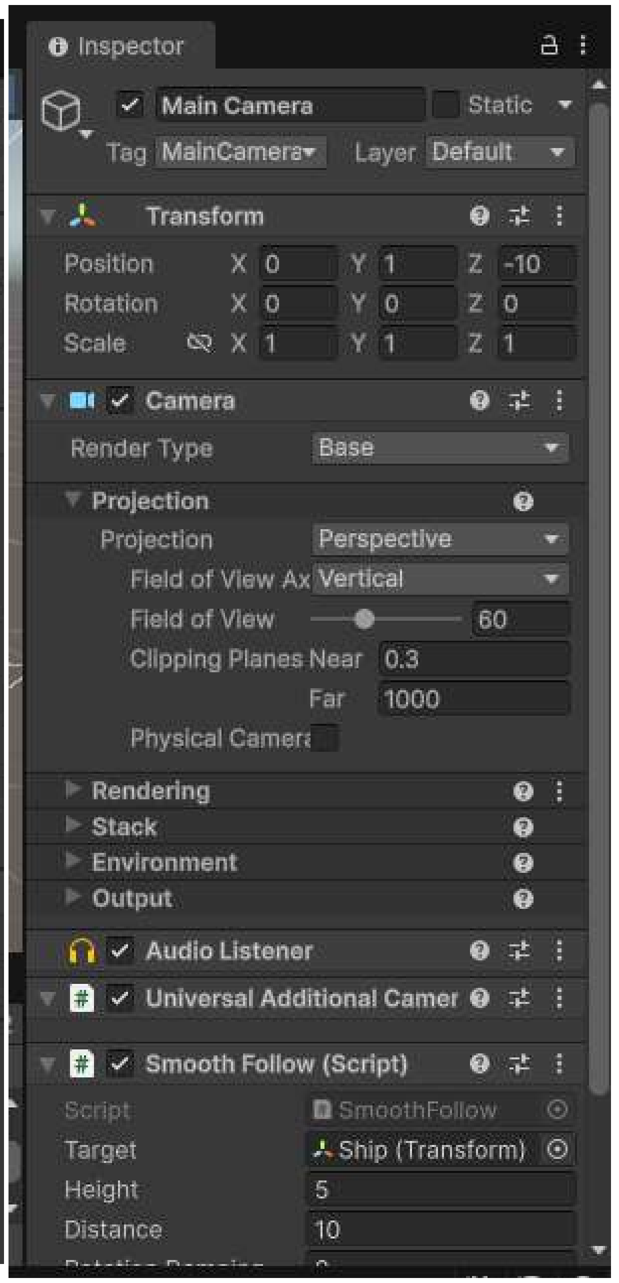
Inspector
Ship — Static
Tag Untagged — Layer Default

Transform
Position X -3.853(Y 0 Z -1.9287
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Box Collider
Edit Collider
Is Trigger ✓
Provides Contacts
Material None (Physics Materia
Center X 0 Y 0 Z 0
Size X 2 Y 1.2 Z 3
Layer Overrides

Ship Thrust (Script)
Script ShipThrust
Speed 5

Add Component



Inspector
Main Camera — Static
Tag MainCamera — Layer Default

Transform
Position X 0 Y 1 Z -10
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

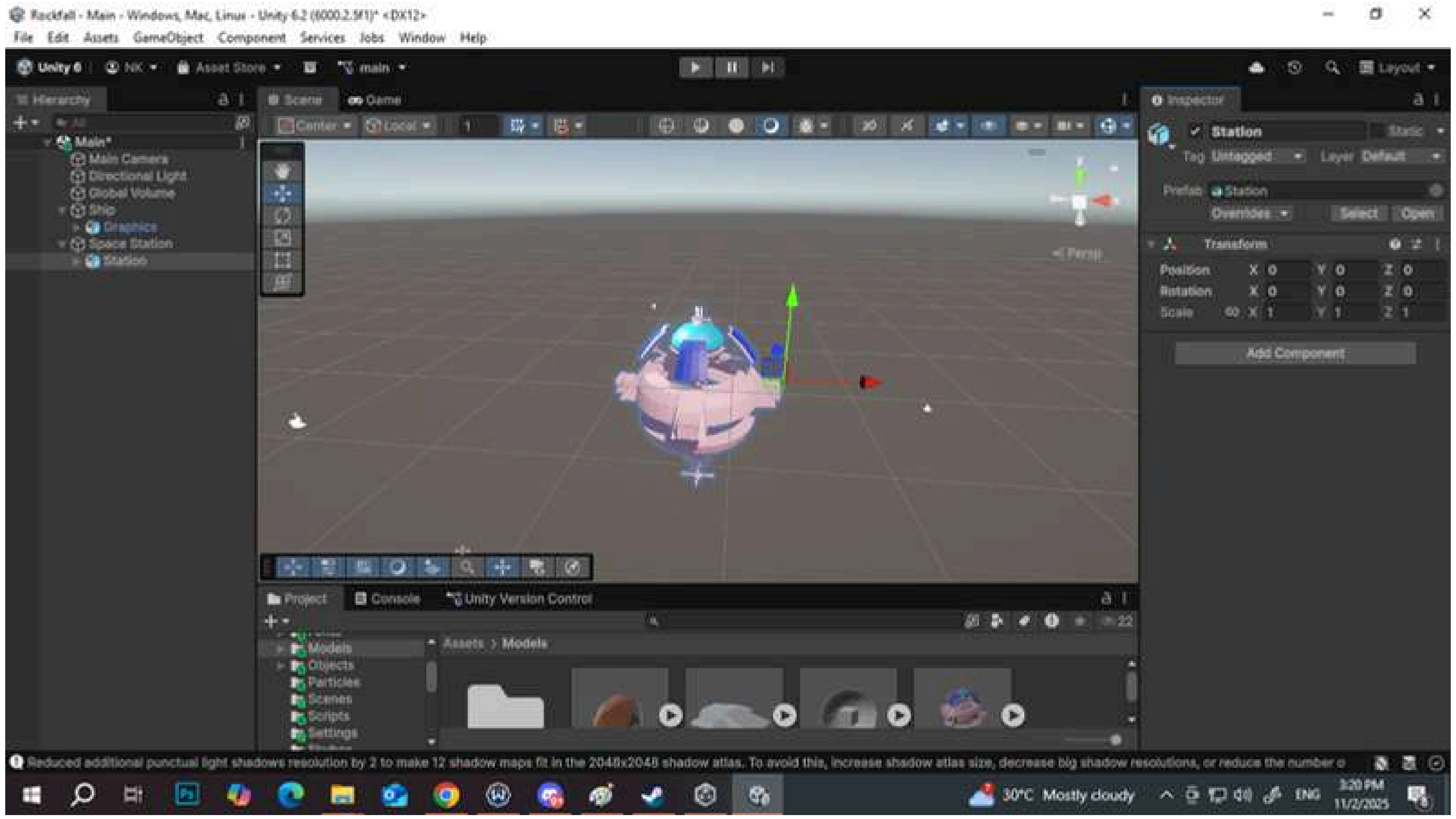
Camera
Render Type Base
Projection
Projection Perspective
Field of View Ax Vertical
Field of View 60
Clipping Planes Near 0.3
Far 1000
Physical Camera
Rendering
Stack
Environment
Output

Audio Listener
Universal Additional Camer

Smooth Follow (Script)
Script SmoothFollow
Target Ship (Transform)
Height 5
Distance 10

STEP 6

Make new game object and import station model and reset position



Reduced additional punctual light shadows resolution by 2 to make 12 shadow maps fit in the 2048x2048 shadow atlas. To avoid this, increase shadow atlas size, decrease big shadow resolutions, or reduce the number o

STEP 7

Adding the Joystick:

- Create the joystick pad UI object, set its size, position (lower-left corner), and assign the Pad sprite to its Source Image property.
- Create the joystick thumb UI object as a child of the pad, center it, set its size, and assign the Thumb sprite to its Source Image property.
- Add the VirtualJoystick.cs script to the joystick pad object.
- Configure the joystick by dragging the Thumb object into the Thumb slot of the VirtualJoystick component.

STEP 8

Input Manager:

- Create a Singleton.cs script (identical to the one used in the 2D game project).
- Create a new empty game object named "Input Manager."
- Add the InputManager.cs script to the "Input Manager" object, which holds a reference to the VirtualJoystick.
- Configure the "Input Manager" by dragging the Joystick object into the Steering slot of the InputManager component.

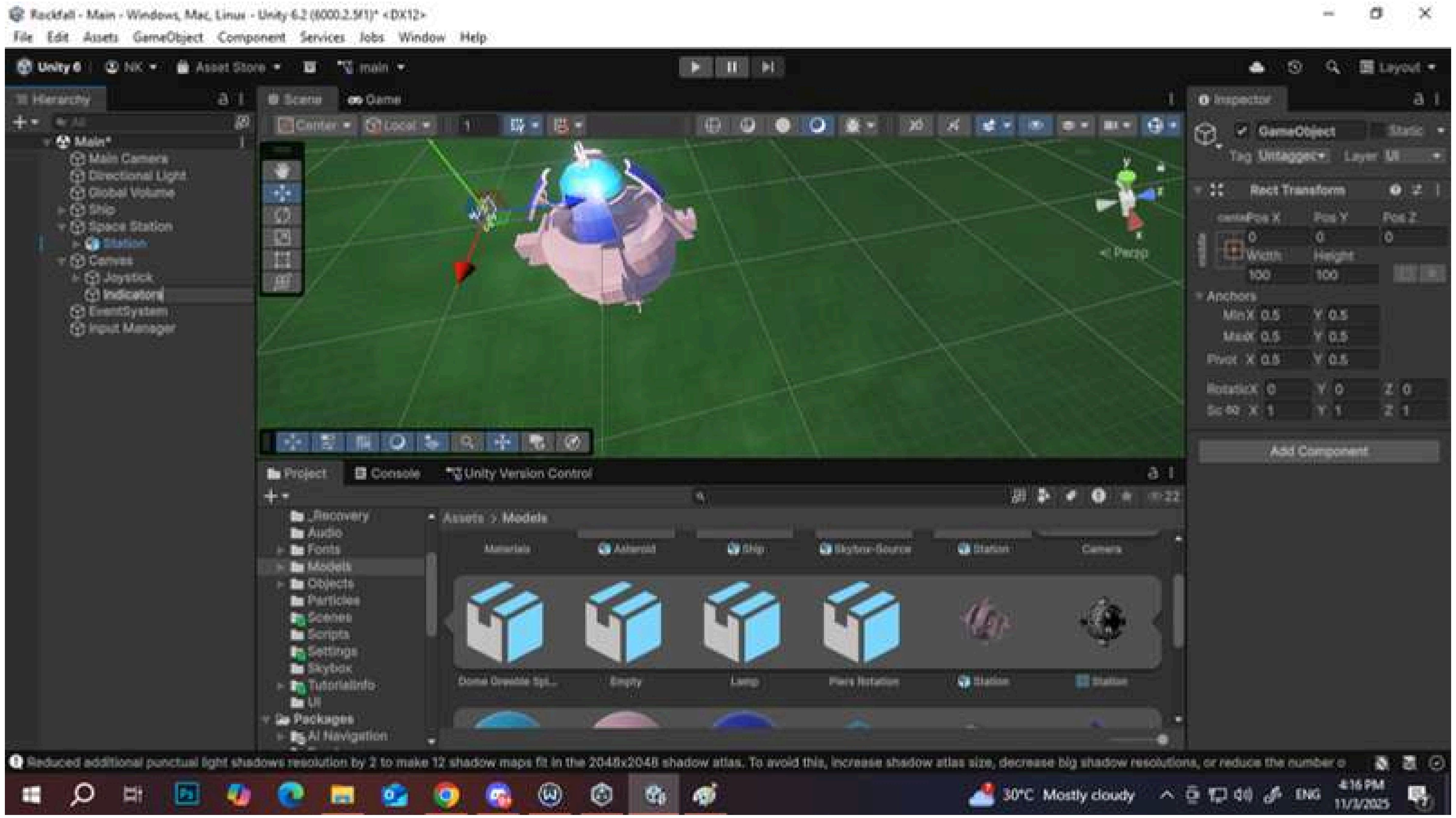
STEP 9

Flight Control

- Implementing Ship Steering:
  - Add the ShipSteering.cs script to the main Ship object.

STEP 10

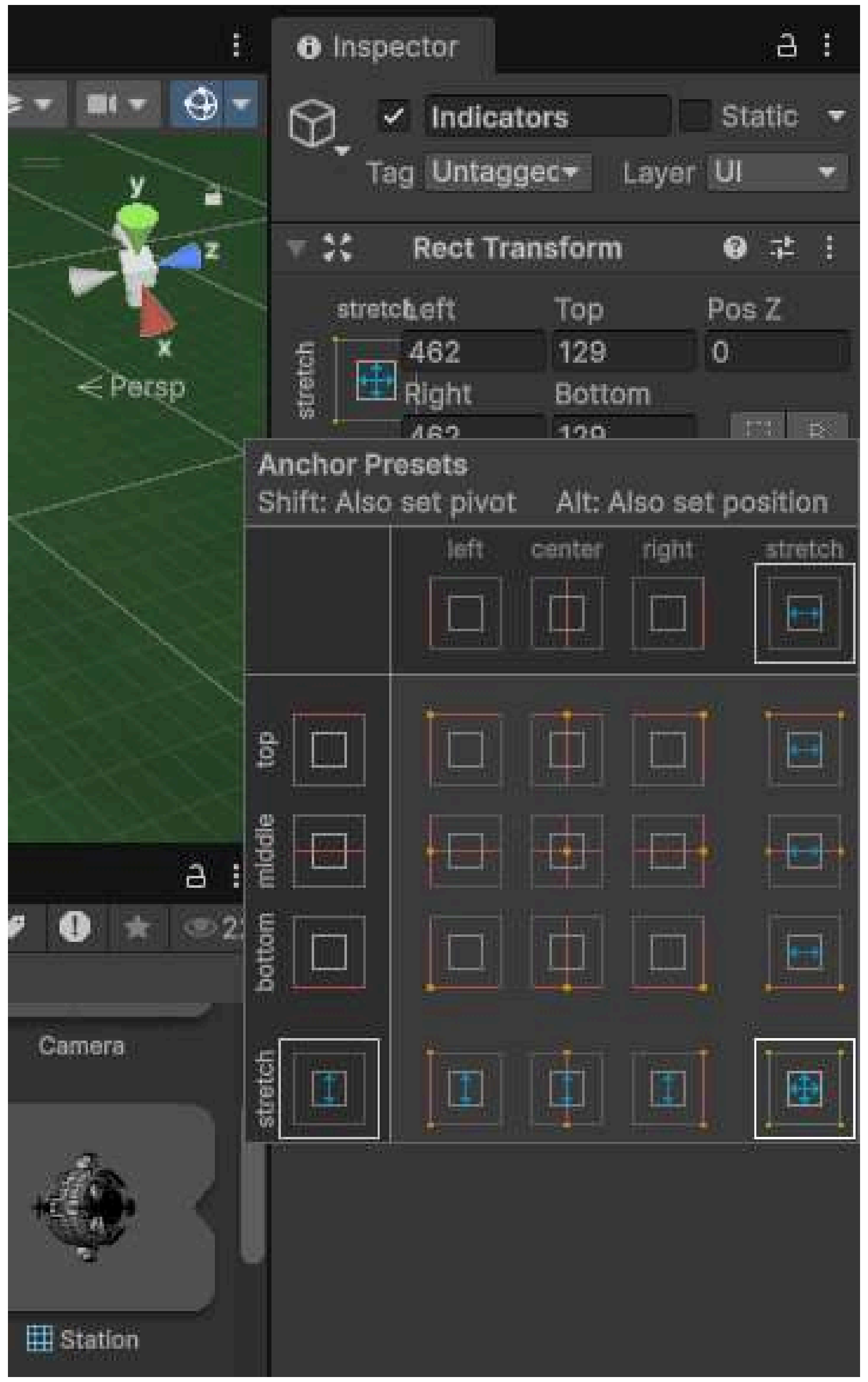
Skipped through flight controls guide, make a new child under Canvas called Indicators



Reduced additional punctual light shadows resolution by 2 to make 12 shadow maps fit in the 2048x2048 shadow atlas. To avoid this, increase shadow atlas size, decrease big shadow resolutions, or reduce the number o

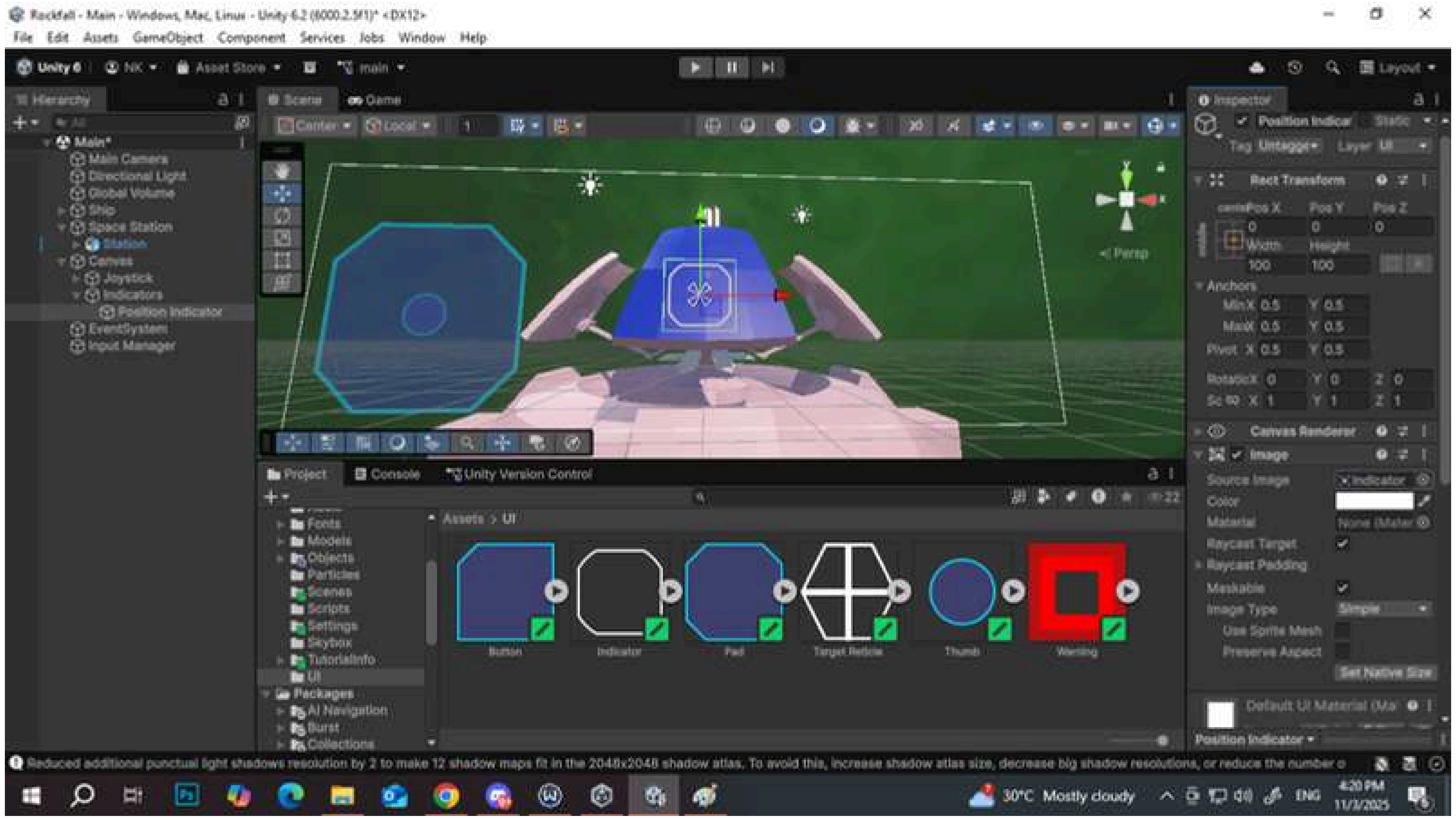
STEP 11

Set Indicators to stretch horizontally and vertically



STEP 12

Make a new Image through GameObject-UI-Image and name it Position Indicator, make it a child of Indicators and add Indicator sprite as its source image


Rockfall - Main - Windows, Mac, Linux - Unity 6.2 (6000.2.5f1)* <DX12>

Inspector: Position Indicator — Tag Untagged — Layer UI

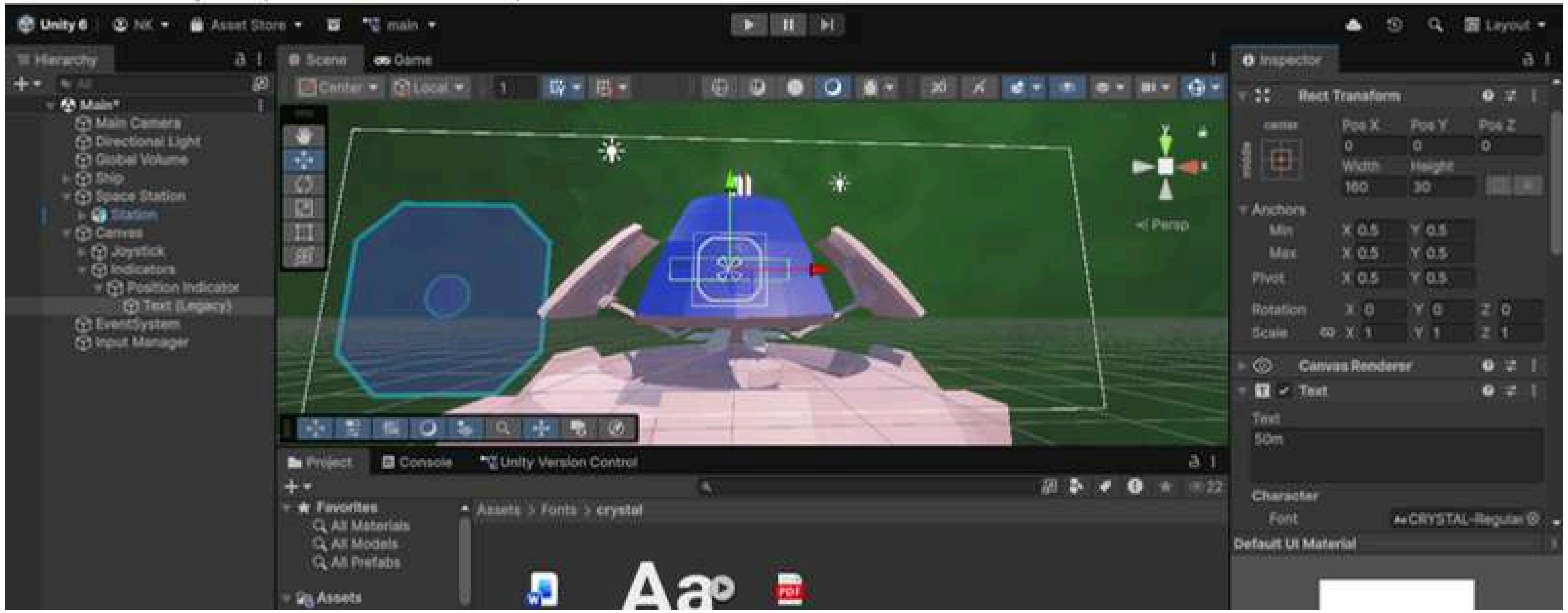
Rect Transform: Pos X 0, Pos Y 0, Pos Z 0; Width 100, Height 100
Anchors: Min X 0.5, Y 0.5; Max X 0.5, Y 0.5; Pivot X 0.5, Y 0.5
Rotation X 0, Y 0, Z 0; Scale X 1, Y 1, Z 1

Image: Source Image Indicator; Material None (Material); Raycast Target ✓; Maskable ✓; Image Type Simple

Reduced additional punctual light shadows resolution by 2 to make 12 shadow maps fit in the 2048x2048 shadow atlas. To avoid this, increase shadow atlas size, decrease big shadow resolutions, or reduce the number o

STEP 13

Add a Text child under Position Indicator, change its position to center middle, X and Y to 0, set text to 50m then add custom font Crystal



- Main*
  - Main Camera
  - Directional Light
  - Global Volume
  - Ship
  - Space Station
    - Station
  - Canvas
    - Joystick
    - Indicators
      - Position Indicator
        - Text (Legacy)
  - EventSystem
  - Input Manager

Rect Transform

center / middle

	Pos X	Pos Y	Pos Z
	0	0	0
	Width	Height	
	160	30	
Anchors			
Min	X 0.5	Y 0.5	
Max	X 0.5	Y 0.5	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Canvas Renderer

Text

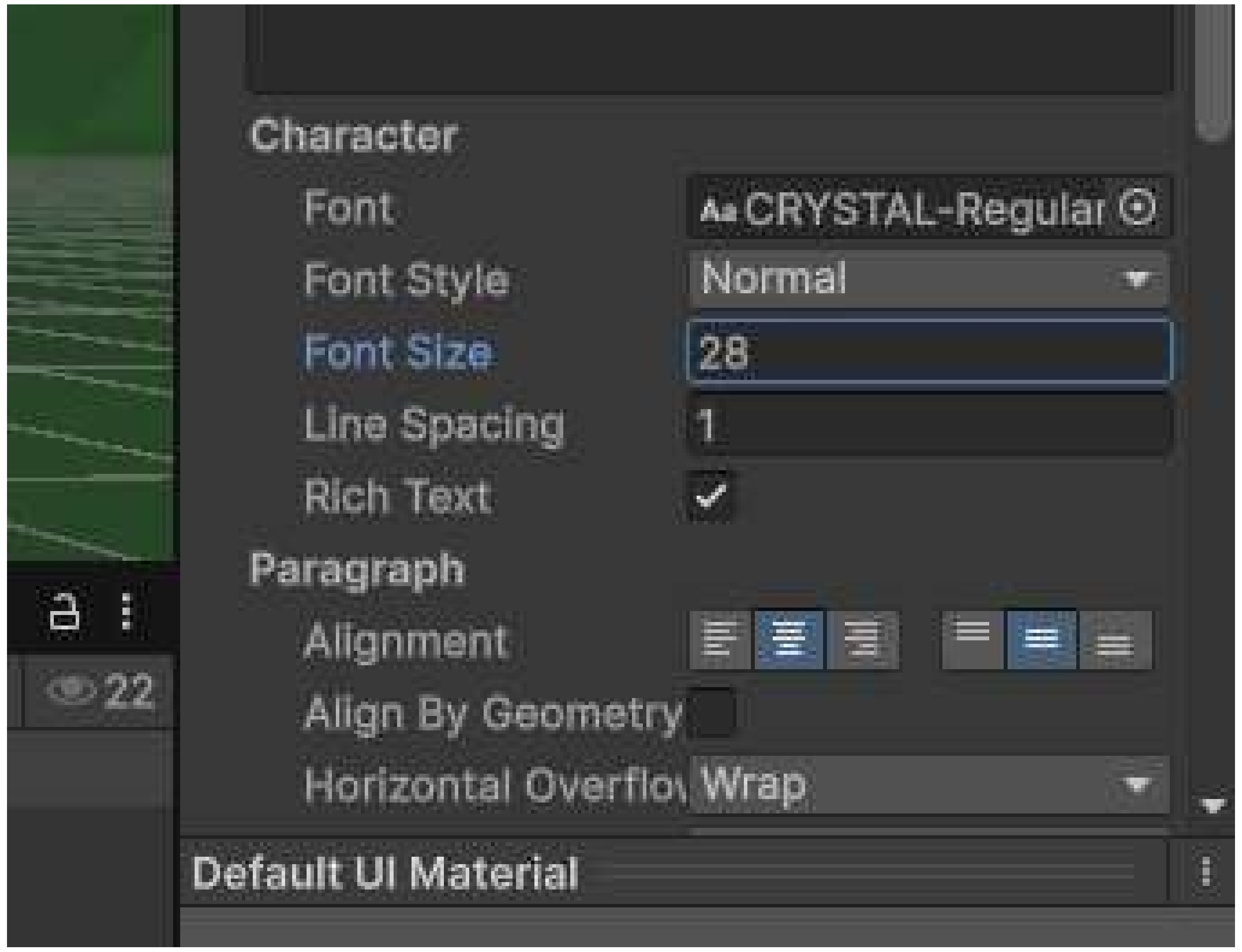
Text
50m

Character
Font CRYSTAL-Regular

Default UI Material

STEP 14

Set font size to 28 and alignment to center both horizontally and vertically



Character

Font: CRYSTAL-Regular

Font Style: Normal

Font Size: 28

Line Spacing: 1

Rich Text: ✓

Paragraph

Alignment

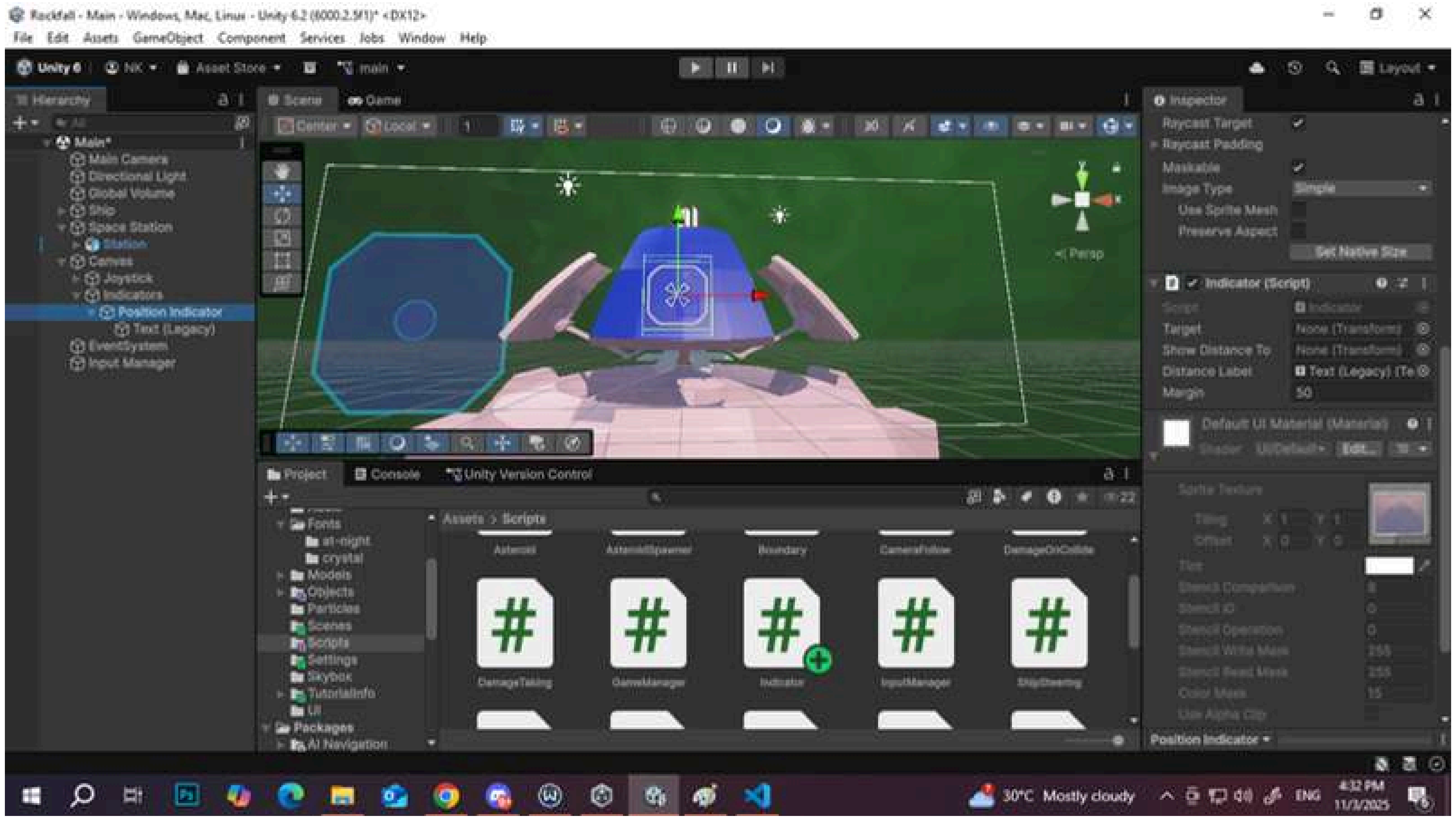
Align By Geometry

Horizontal Overflow: Wrap

Default UI Material

STEP 15

Make new script Indicator.cs then add it to position indicator and drag the text to the distance label area



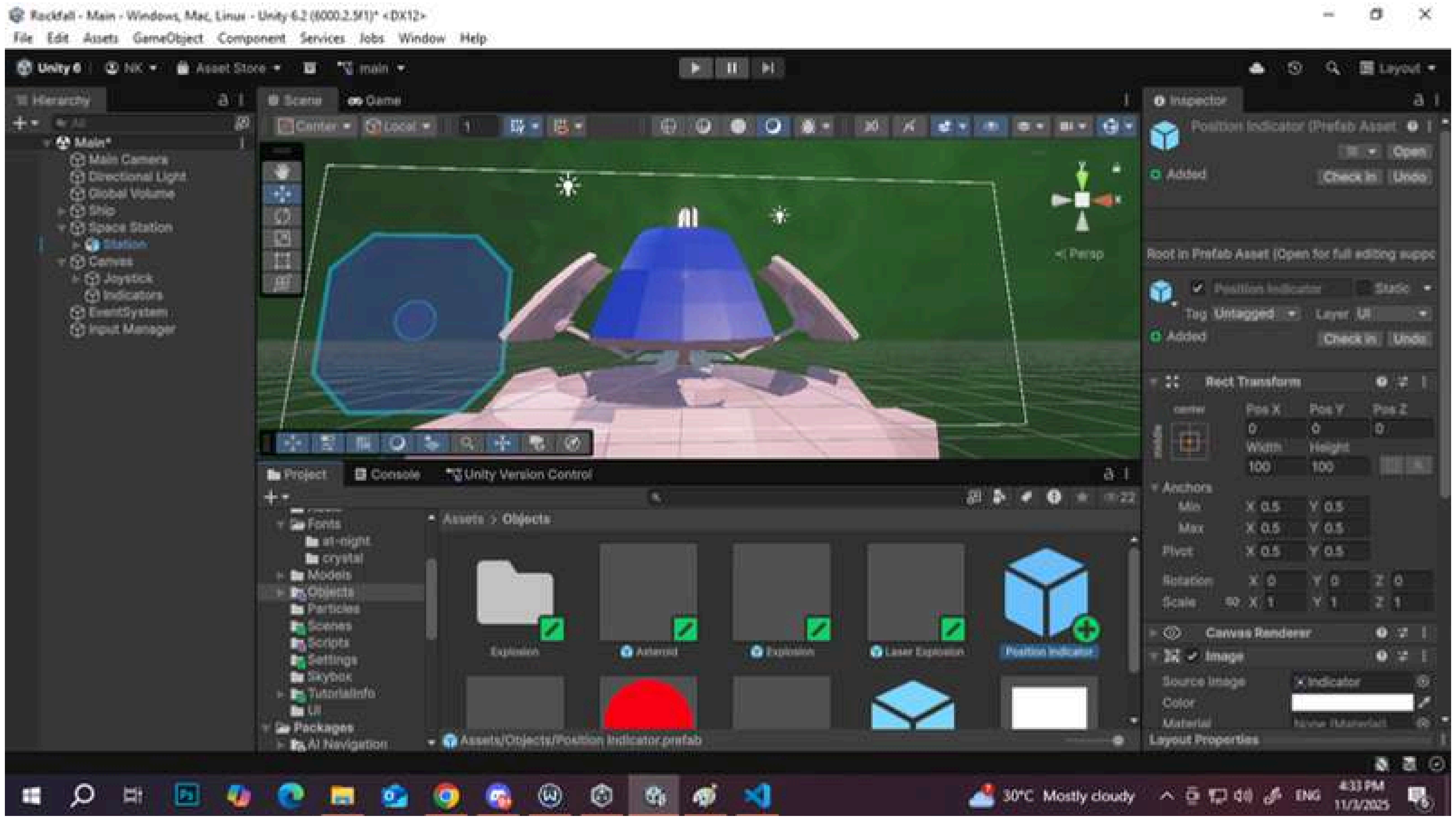
Rockfall - Main - Windows, Mac, Linux - Unity 6.2 (6000.2.5f1)* <DX12>

Indicator (Script)

Script	Indicator
Target	None (Transform)
Show Distance To	None (Transform)
Distance Label	Text (Legacy) (Te
Margin	50

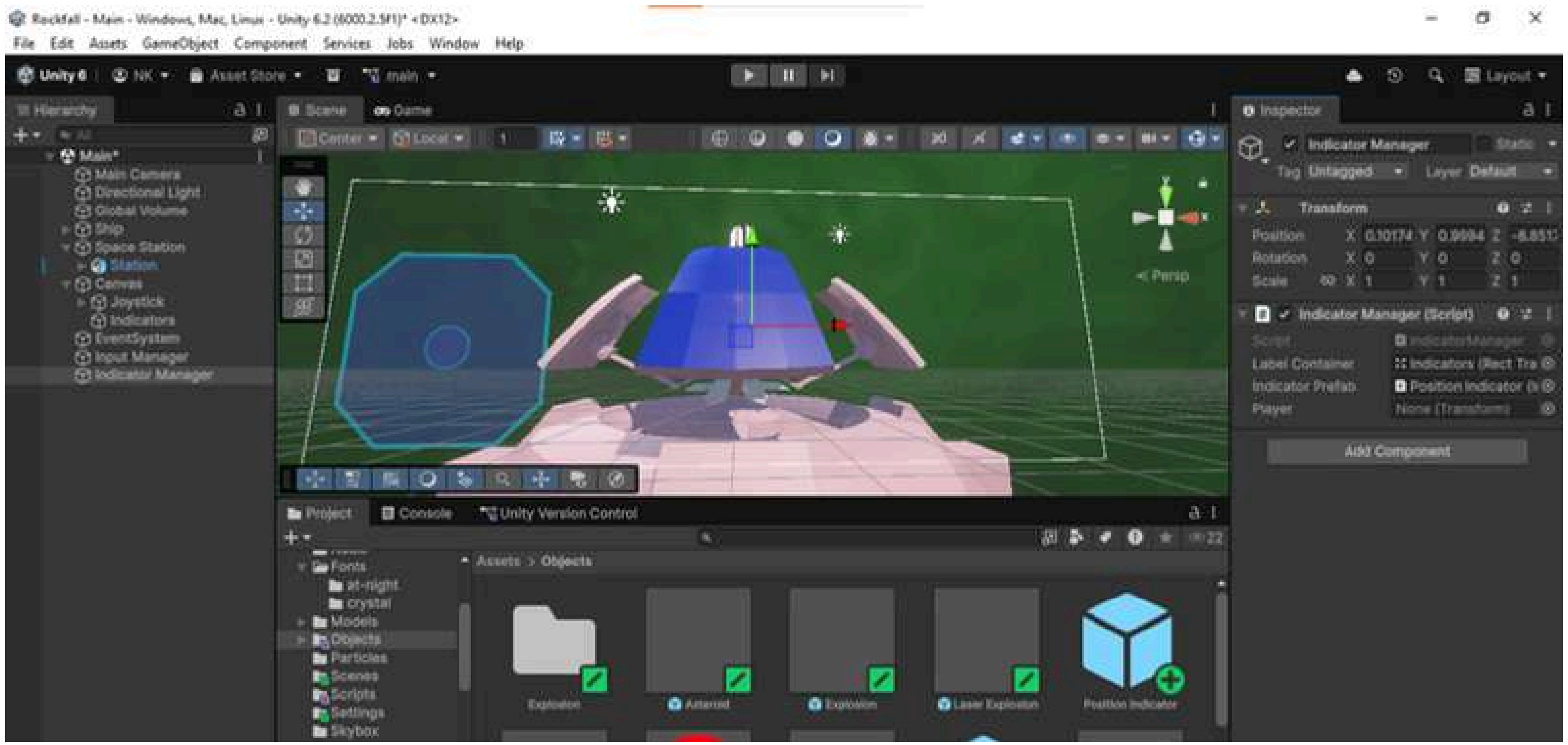
STEP 16

Turn the Position Indicator into a prefab and delete it from the scene sheesh



STEP 17

Create new game object Indicator Manager, add Indicator Manager script, drag Indicators object to label container and Position Indicator to indicator prefab



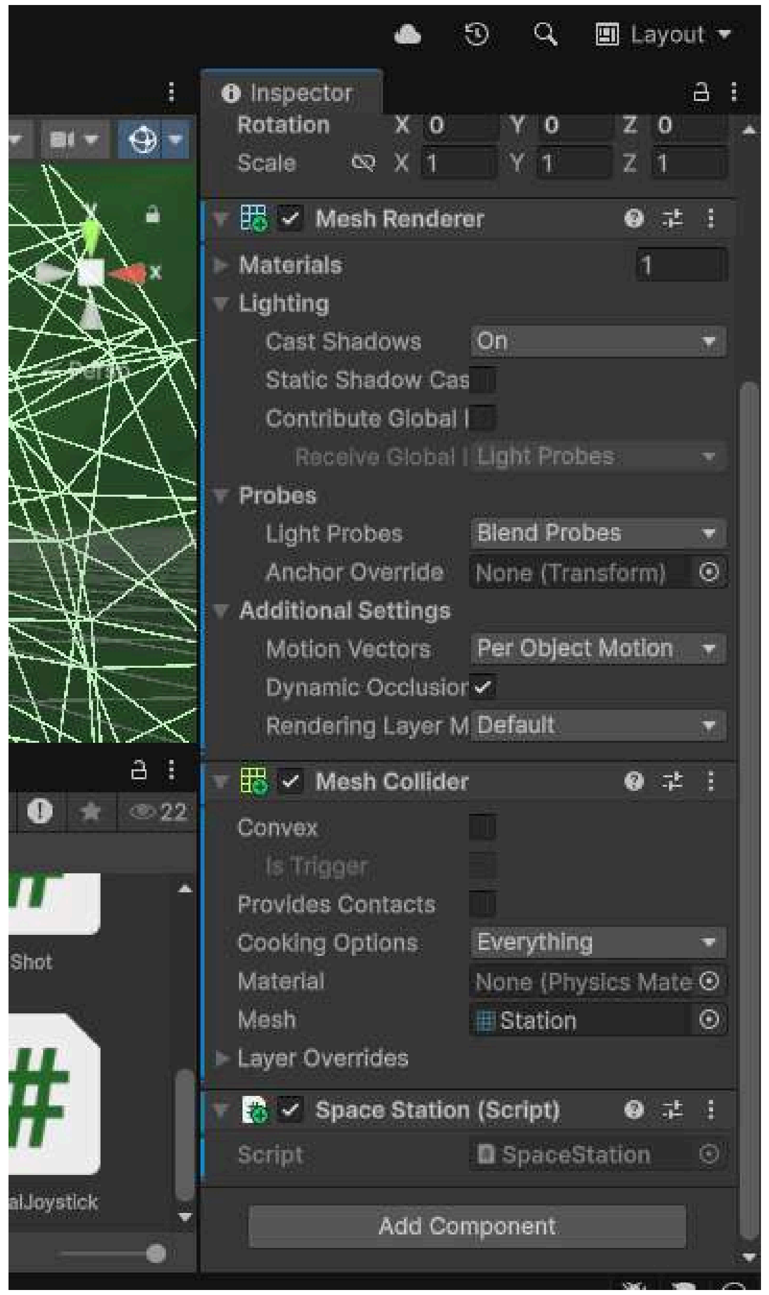
Indicator Manager
Tag Untagged Layer Default

Transform
Position X 0.10174 Y 0.9994 Z -6.851
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Indicator Manager (Script)
Script IndicatorManager
Label Container Indicators (Rect Tra
Indicator Prefab Position Indicator (I
Player None (Transform)

STEP 18

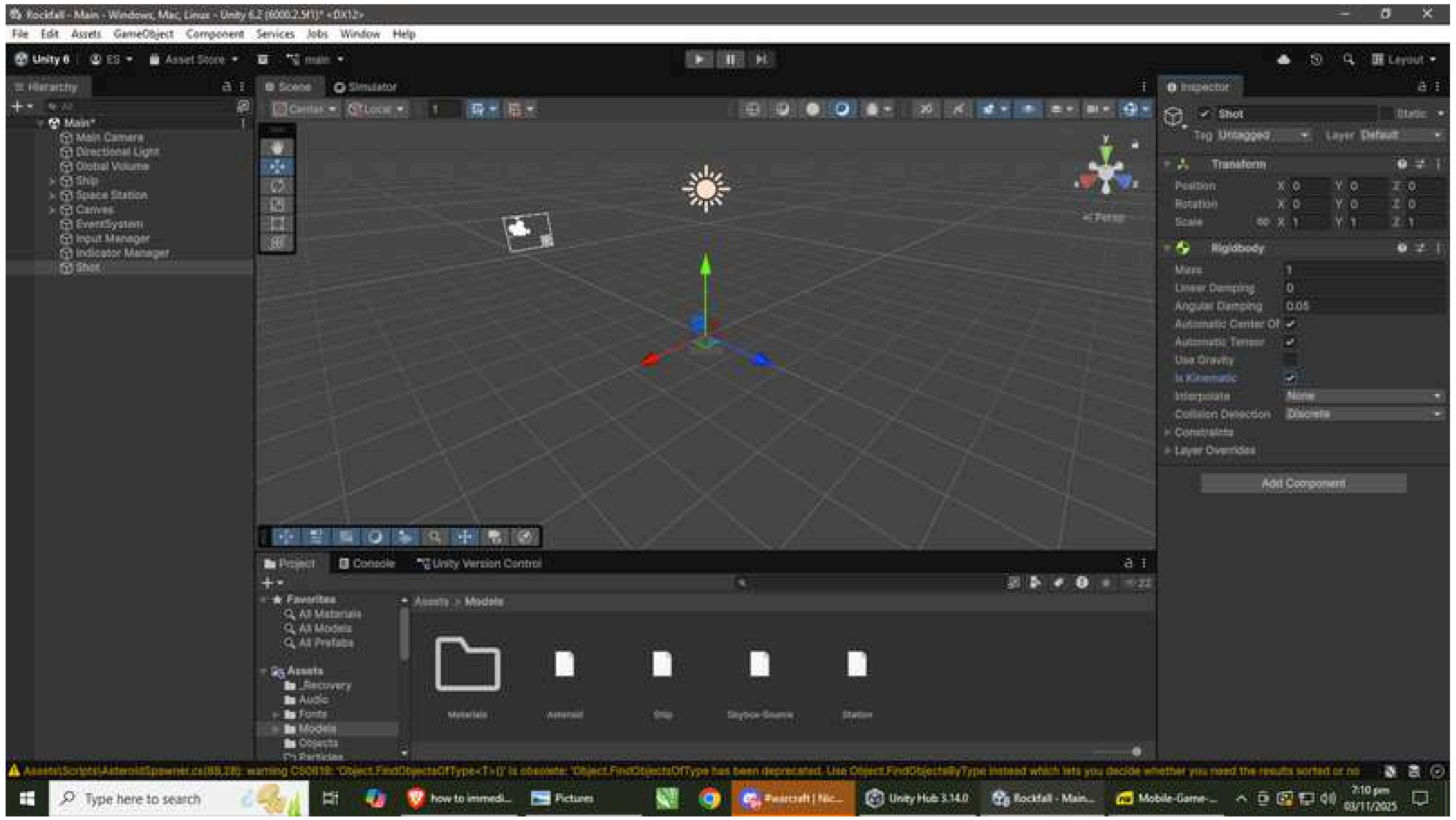
Add Space Station script to Station



```
Inspector
Rotation   X 0   Y 0   Z 0
Scale      X 1   Y 1   Z 1
Mesh Renderer
  Materials                    1
  Lighting
    Cast Shadows               On
    Static Shadow Cas          [ ]
    Contribute Global I        [ ]
    Receive Global I           Light Probes
  Probes
    Light Probes               Blend Probes
    Anchor Override            None (Transform)
  Additional Settings
    Motion Vectors             Per Object Motion
    Dynamic Occlusior          [✓]
    Rendering Layer M          Default
Mesh Collider
  Convex                       [ ]
    Is Trigger                 [ ]
  Provides Contacts            [ ]
  Cooking Options              Everything
  Material                     None (Physics Mate
  Mesh                         Station
  Layer Overrides
Space Station (Script)
  Script                       SpaceStation
Add Component
```

STEP 19

Create a new game object named “Shot” and add a rigidbody component to the object. Then make sure Use Gravity is turned off, and Is Kinematic is turned on.



Inspector

Shot

Tag Untagged · Layer Default

Transform

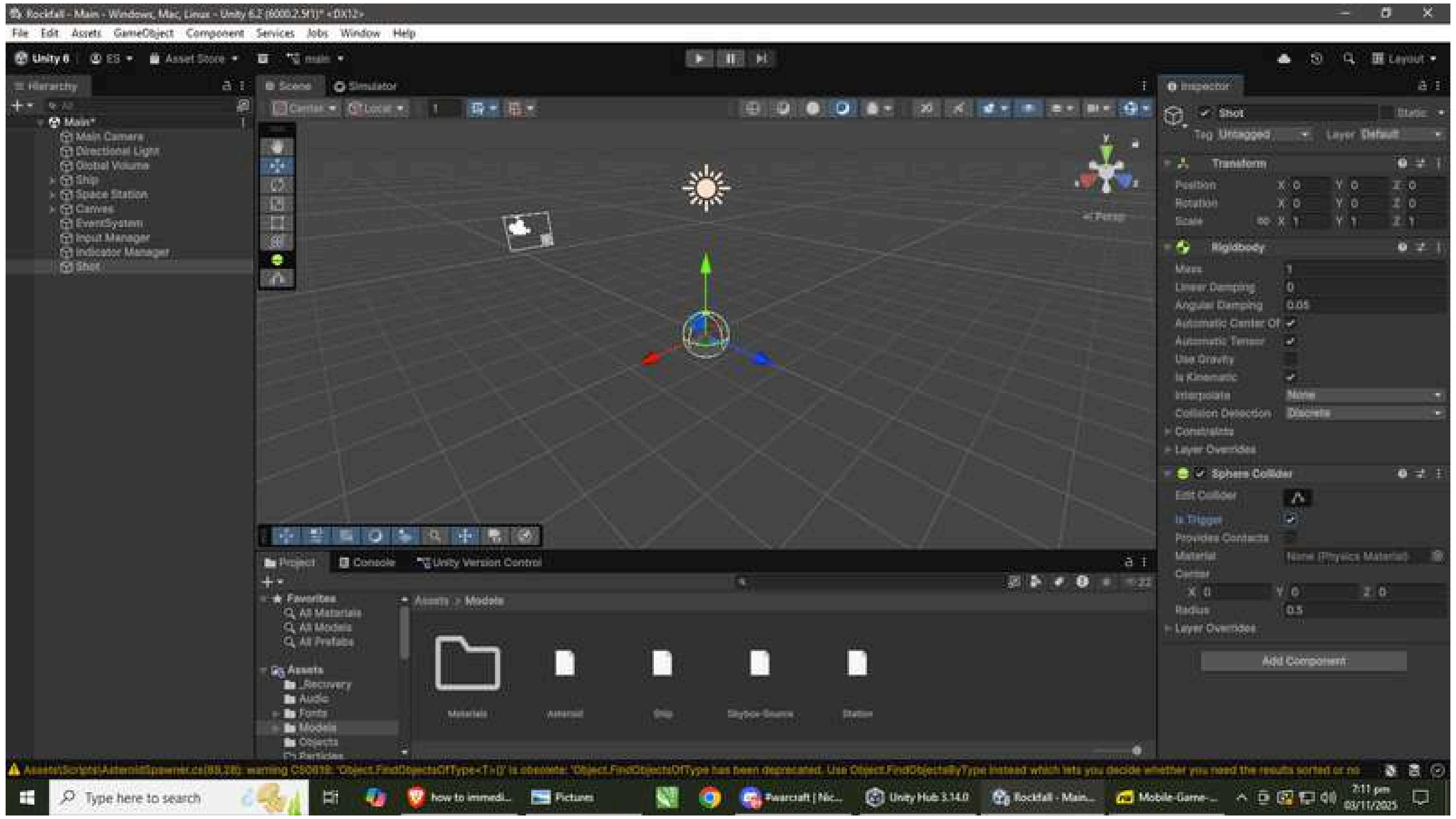
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Rigidbody

Mass	1
Linear Damping	0
Angular Damping	0.05
Automatic Center Of	✓
Automatic Tensor	✓
Use Gravity	
Is Kinematic	✓
Interpolate	None
Collision Detection	Discrete

STEP 20

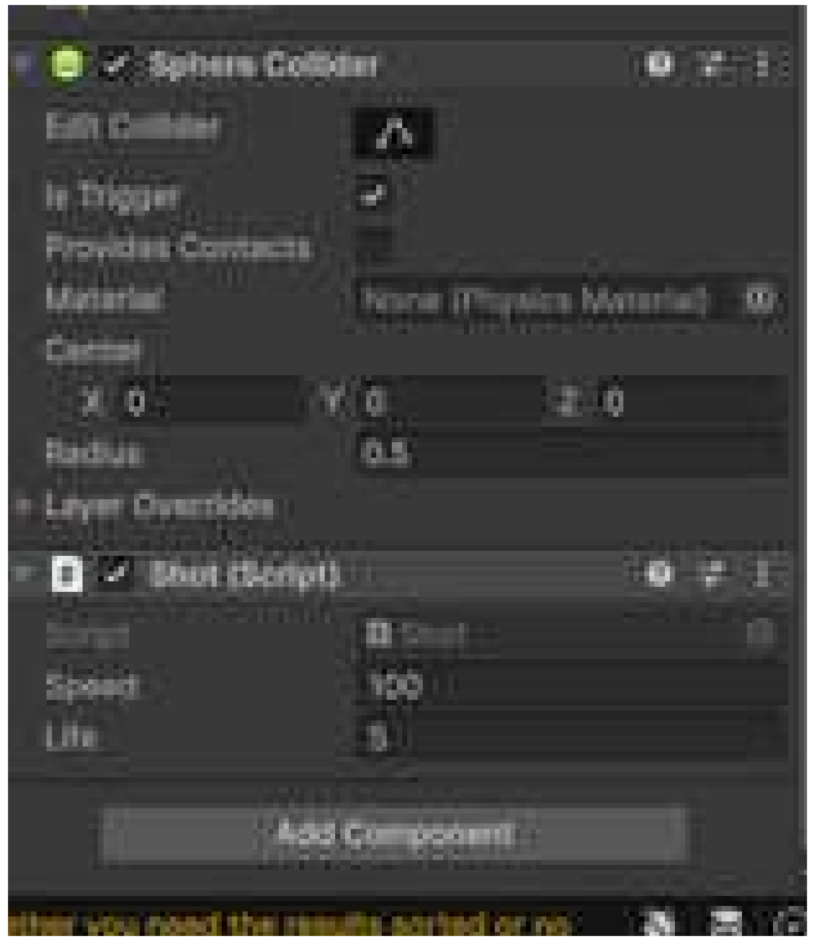
Add a sphere collider to the object. Set its radius to 0.5, and ensure that its center is (0,0,0). The Is Trigger setting should be turned on.



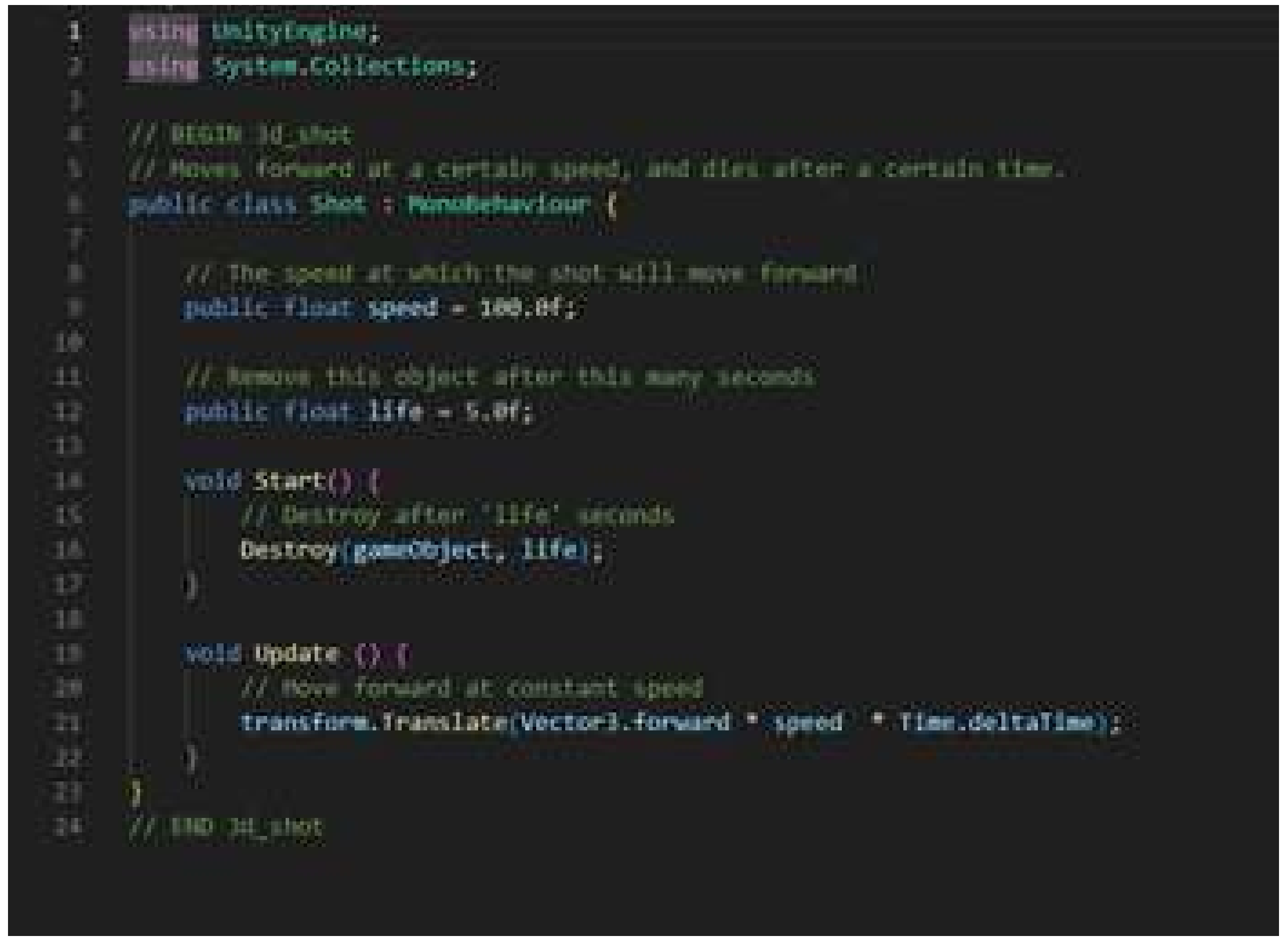
```
Inspector
✓ Shot                                   Static
Tag Untagged          Layer Default
Transform
Position    X 0    Y 0    Z 0
Rotation    X 0    Y 0    Z 0
Scale       X 1    Y 1    Z 1
Rigidbody
Mass                    1
Linear Damping          0
Angular Damping         0.05
Automatic Center Of     ✓
Automatic Tensor        ✓
Use Gravity
Is Kinematic            ✓
Interpolate             None
Collision Detection     Discrete
Constraints
Layer Overrides
✓ Sphere Collider
Edit Collider
Is Trigger              ✓
Provides Contacts
Material                None (Physics Material)
Center
  X 0    Y 0    Z 0
Radius                  0.5
Layer Overrides
Add Component
```

STEP 21

Add a new C# script called Shot to the object.



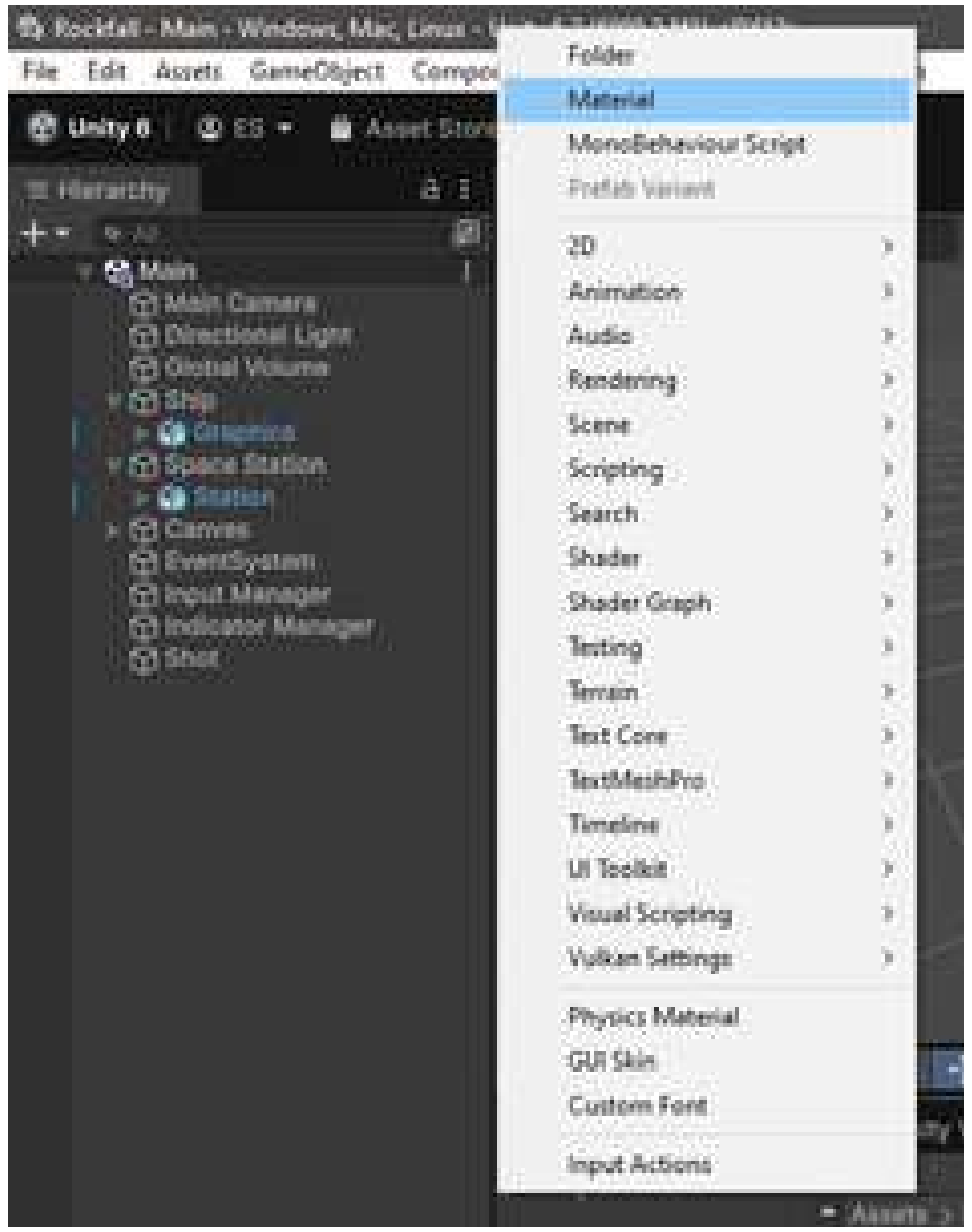
```
Sphere Collider
Edit Collider
Is Trigger          ✓
Provides Contacts
Material            None (Physics Material)
Center
  X 0      Y 0      Z 0
Radius              0.5
Layer Overrides
Shot (Script)
Script              Shot
Speed               100
Life                5
```

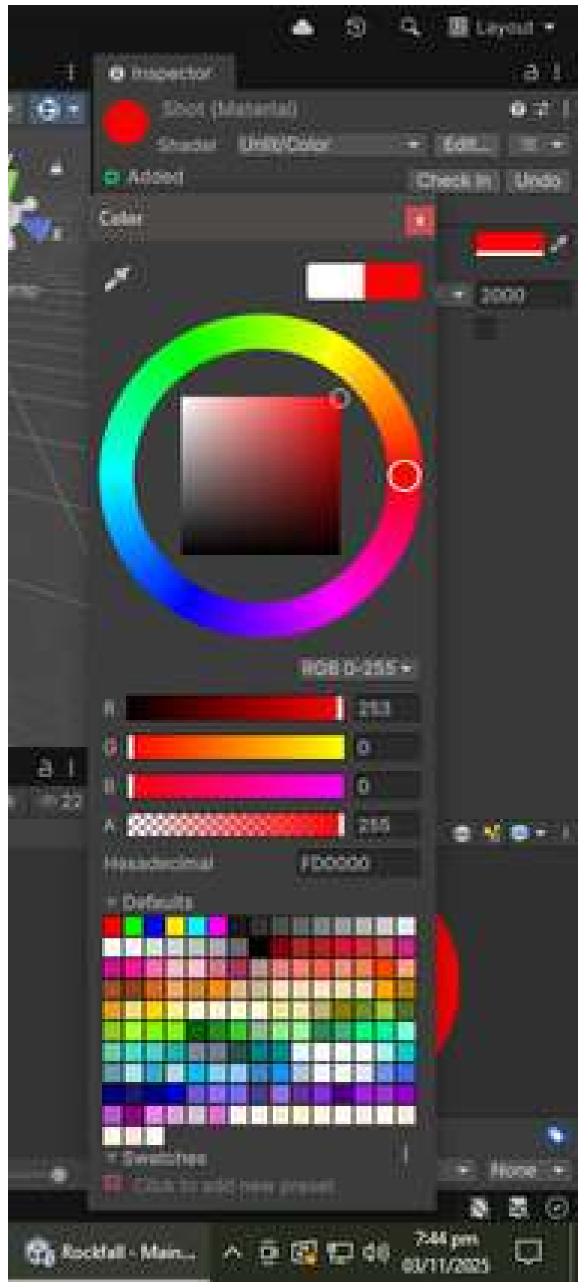


```
using UnityEngine;
using System.Collections;

// BEGIN 3d_shot
// Moves forward at a certain speed, and dies after a certain time.
public class Shot : MonoBehaviour {

    // The speed at which the shot will move forward
    public float speed = 100.0f;

    // Remove this object after this many seconds
    public float life = 5.0f;

    void Start() {
        // Destroy after 'life' seconds
        Destroy(gameObject, life);
    }

    void Update () {
        // Move forward at constant speed
        transform.Translate(Vector3.forward * speed  * Time.deltaTime);
    }
}
// END 3d_shot
```

STEP 22

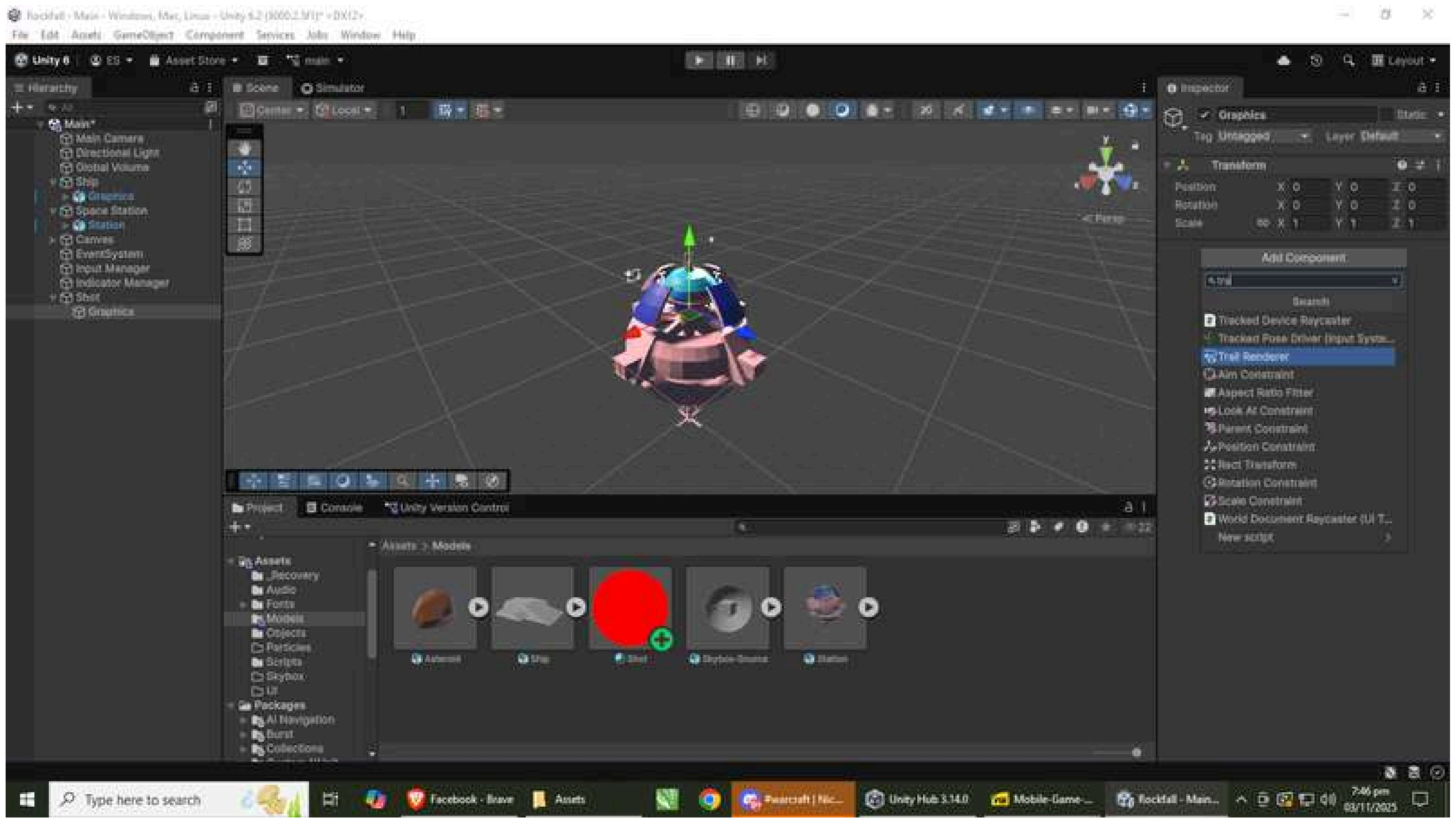
Create a new material named Shot and set its colors.





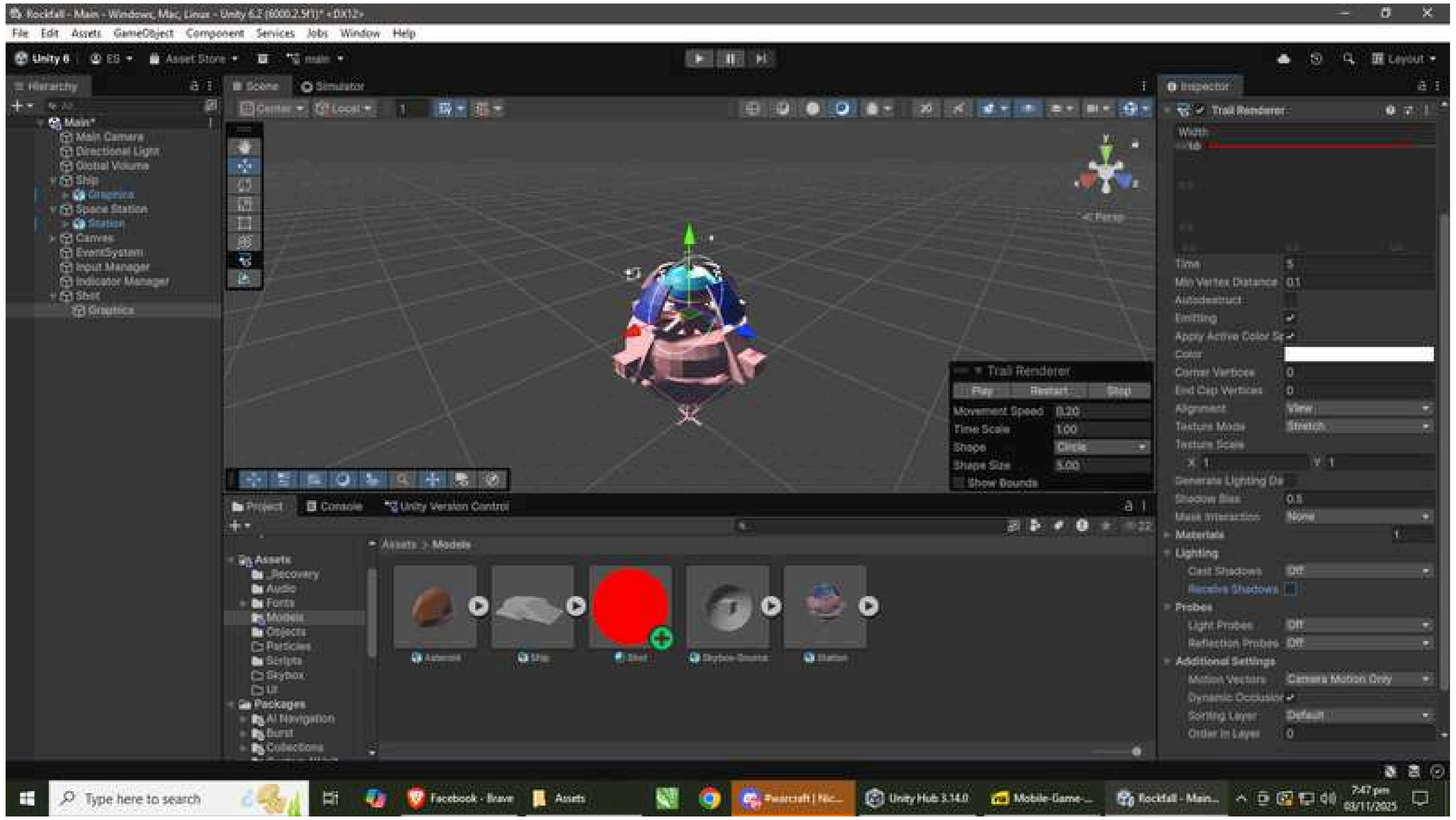
STEP 23

Create a new empty object, and name it “Graphics”. Make it a child of the Shot object, and set its position to (0,0,0). Add a new Trail Renderer component to the Graphics object.



STEP 24

Once it’s added, the Cast Shadows, Receive Shadows, and Use Light Probes should all be turned off. Next, set Time to 0.05, and Width to 0.2



Trail Renderer

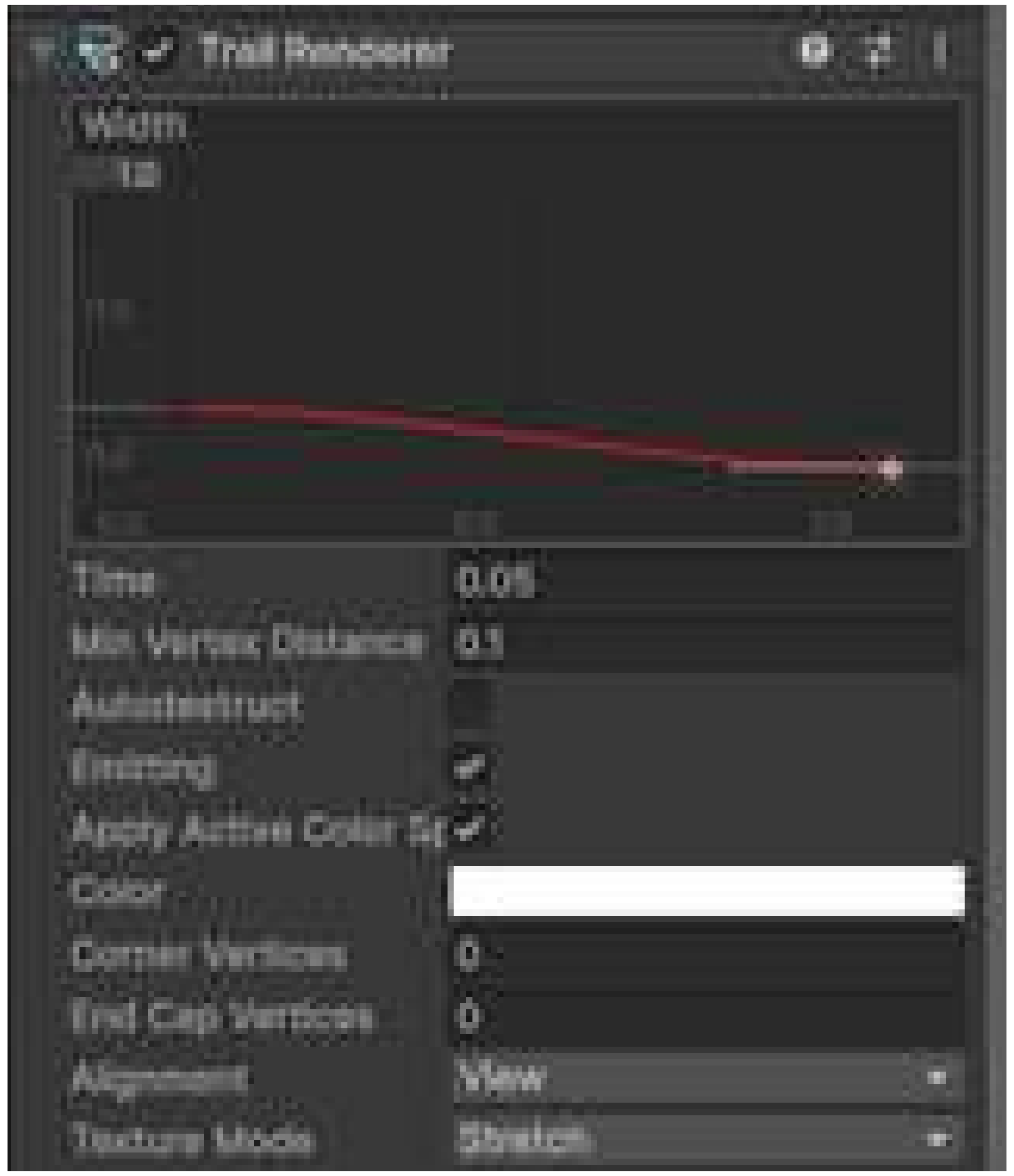
Width	
Time	5
Min Vertex Distance	0.1
Autodestruct	
Emitting	✓
Apply Active Color Sp	✓
Color	
Corner Vertices	0
End Cap Vertices	0
Alignment	View
Texture Mode	Stretch
Texture Scale	X 1 Y 1
Generate Lighting Da	
Shadow Bias	0.5
Mask Interaction	None
Materials	1
Lighting	
Cast Shadows	Off
Receive Shadows	
Probes	
Light Probes	Off
Reflection Probes	Off
Additional Settings	
Motion Vectors	Camera Motion Only
Dynamic Occlusion	✓
Sorting Layer	Default
Order In Layer	0

Trail Renderer

Movement Speed	0.20
Time Scale	1.00
Shape	Circle
Shape Size	5.00
Show Bounds	

STEP 25

Double-click in the curve view (below the Width field), and a new control point will appear. Drag this new control point to the bottom-right of the curve view.


Trail Renderer

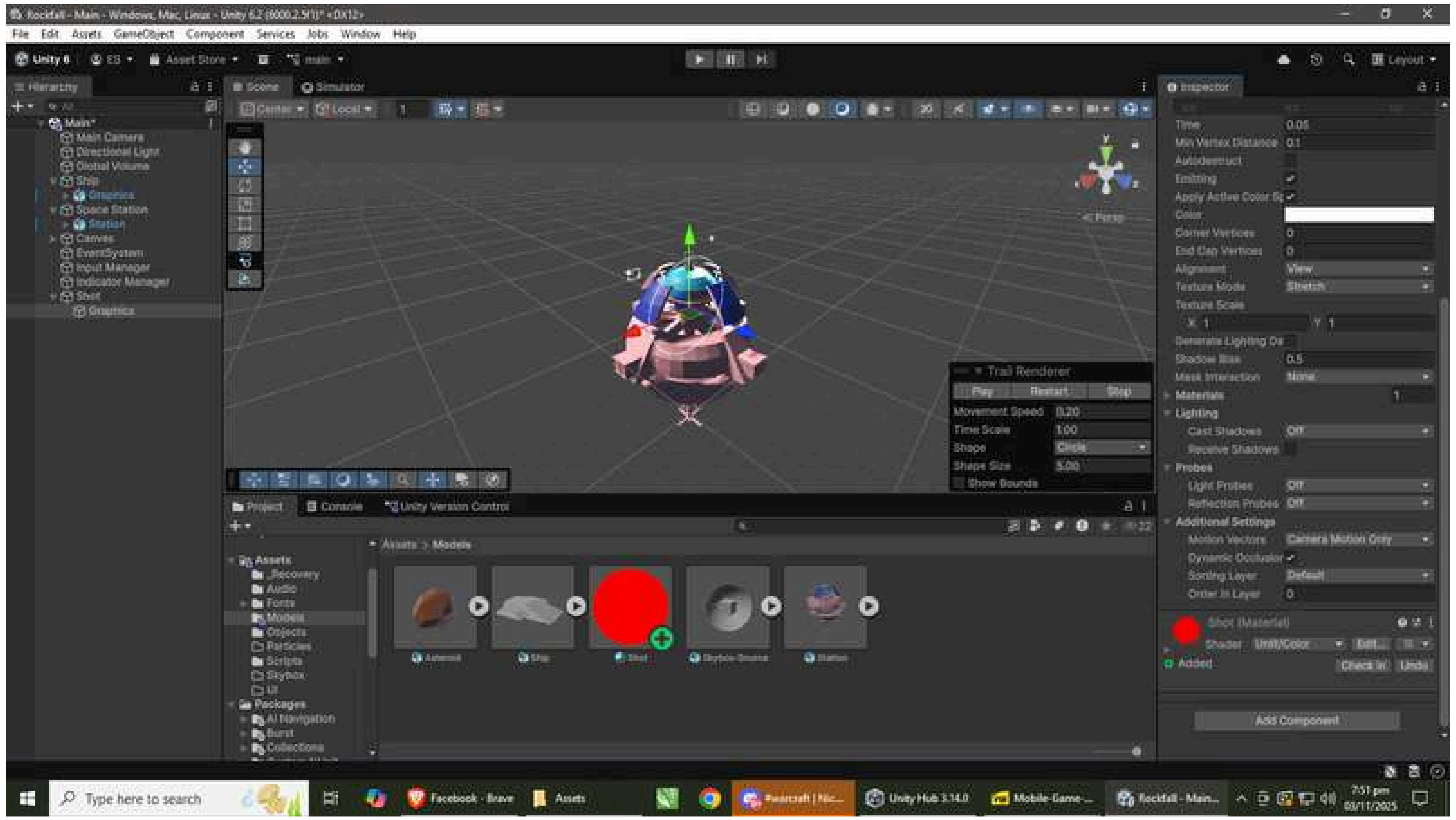
Width

1.0

Time	0.05
Min Vertex Distance	0.1
Autodestruct	
Emitting	✓
Apply Active Color Sp	✓
Color	
Corner Vertices	0
End Cap Vertices	0
Alignment	View
Texture Mode	Stretch

STEP 26

Open the list of Materials, and drag in the Shot material that you just created.



Inspector

Time	0.05
Min Vertex Distance	0.1
Autodestruct	
Emitting	✓
Apply Active Color Sp	✓
Color	
Corner Vertices	0
End Cap Vertices	0
Alignment	View
Texture Mode	Stretch
Texture Scale	X 1 Y 1
Generate Lighting Da	
Shadow Bias	0.5
Mask Interaction	None
Materials	1
Lighting	
Cast Shadows	Off
Receive Shadows	
Probes	
Light Probes	Off
Reflection Probes	Off
Additional Settings	
Motion Vectors	Camera Motion Only
Dynamic Occlusion	✓
Sorting Layer	Default
Order in Layer	0

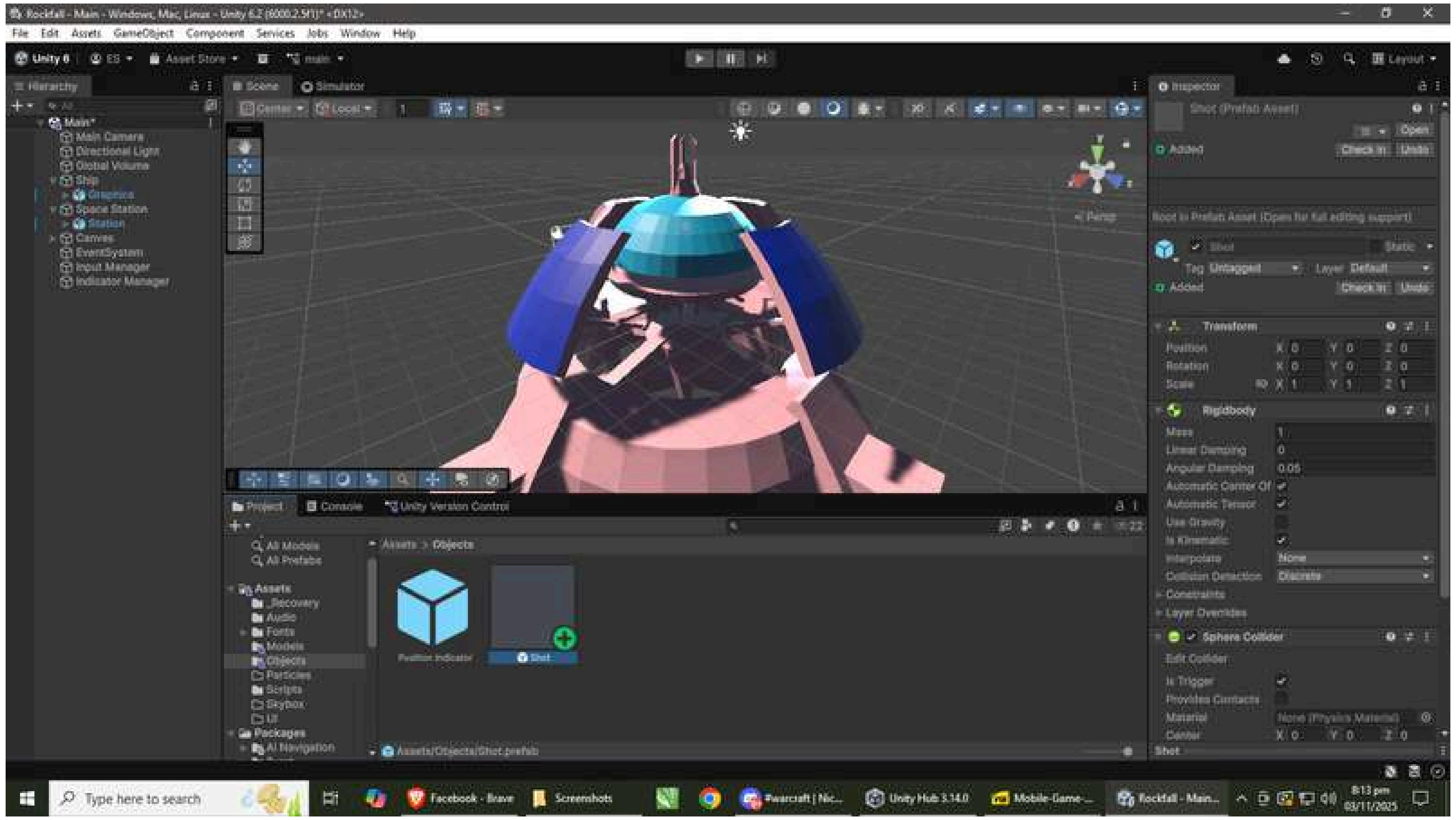
Shot (Material)
Shader Unlit/Color

Trail Renderer

Movement Speed	0.20
Time Scale	1.00
Shape	Circle
Shape Size	5.00
Show Bounds	

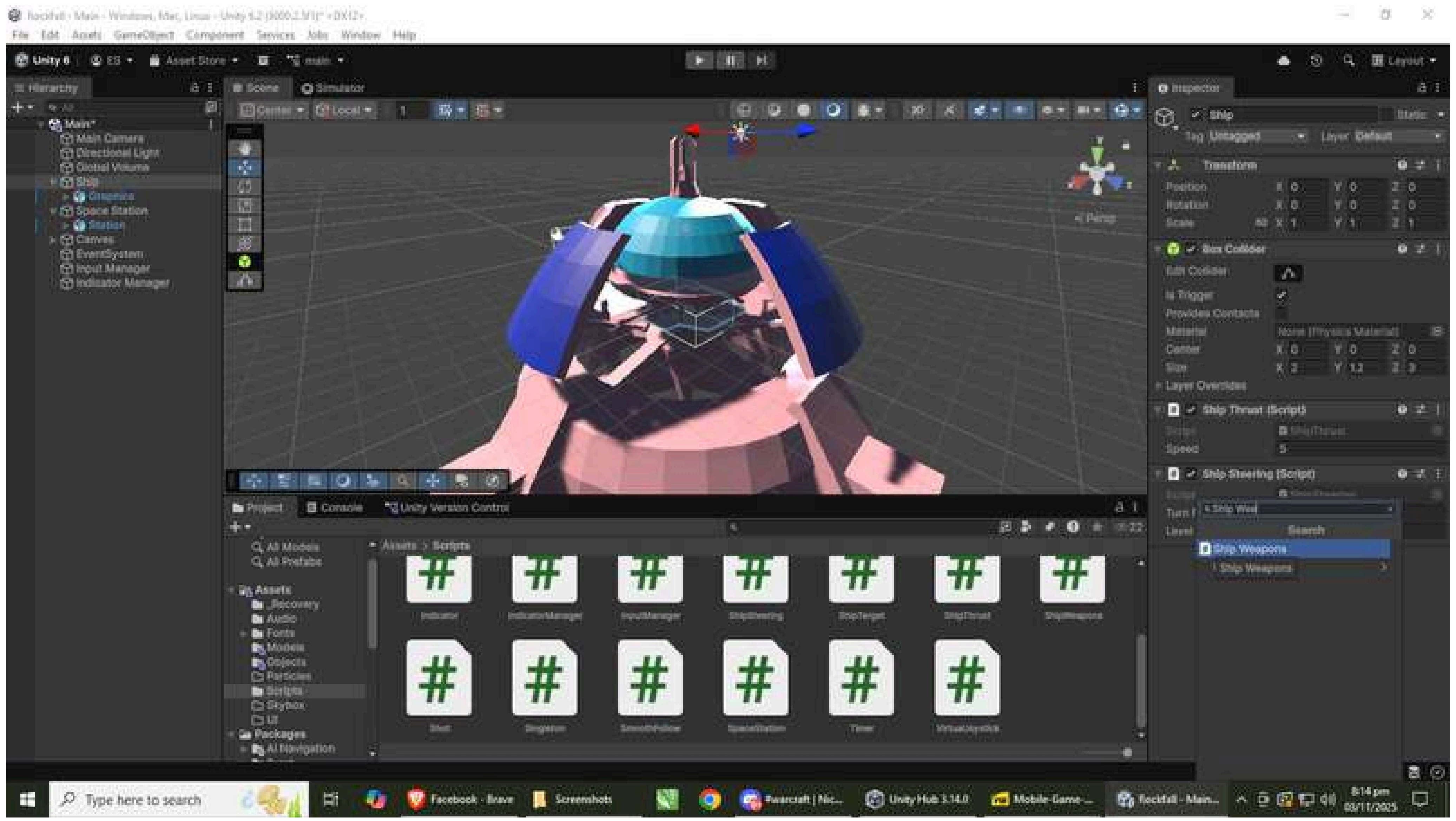
STEP 27

Drag the Shot object from the scene into the Objects folder. This will turn the Shot into a prefab. Delete the Shot from the scene.



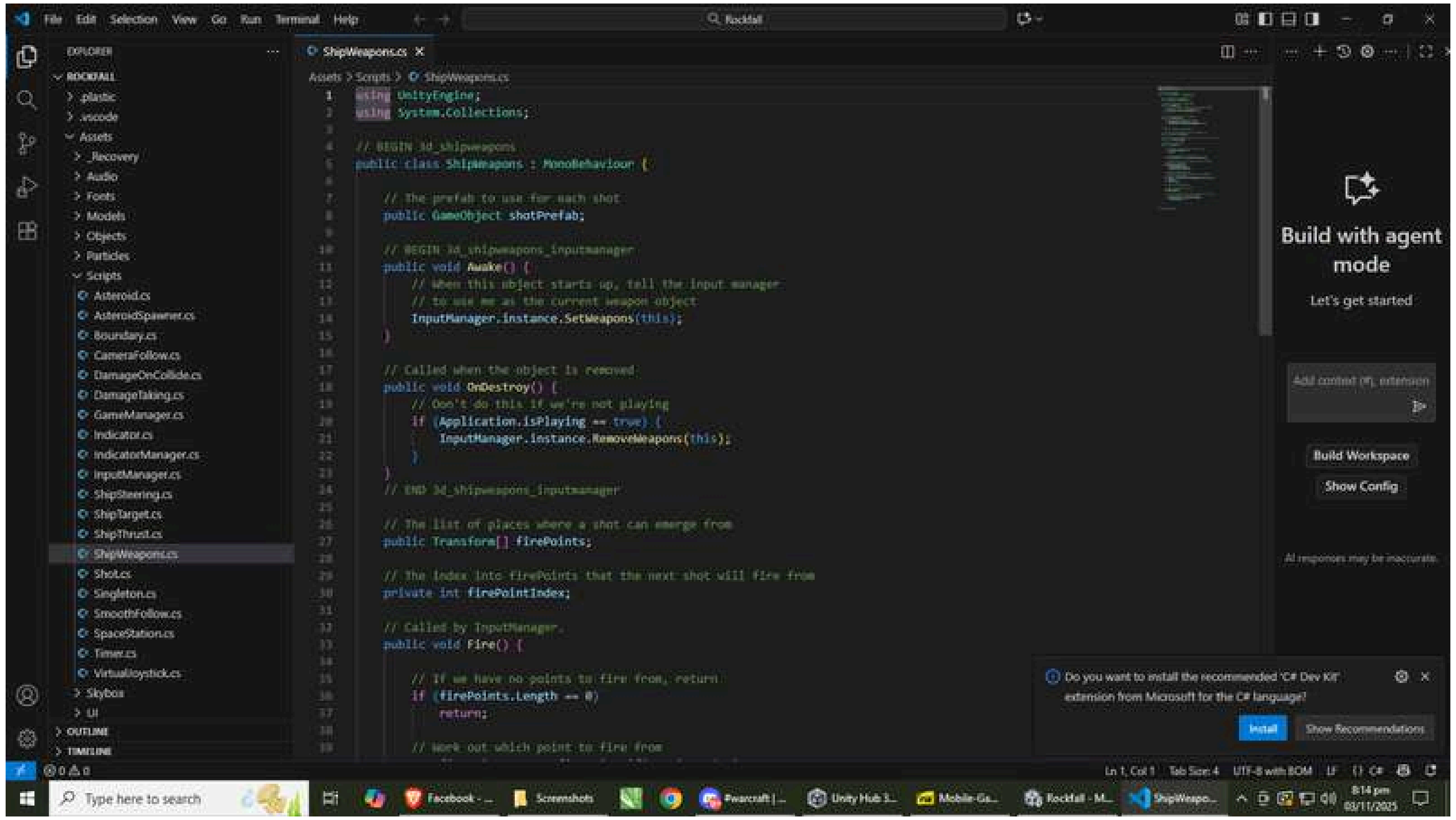
STEP 28

Select the Ship, add a new C# script called ShipWeapons.cs,



STEP 29

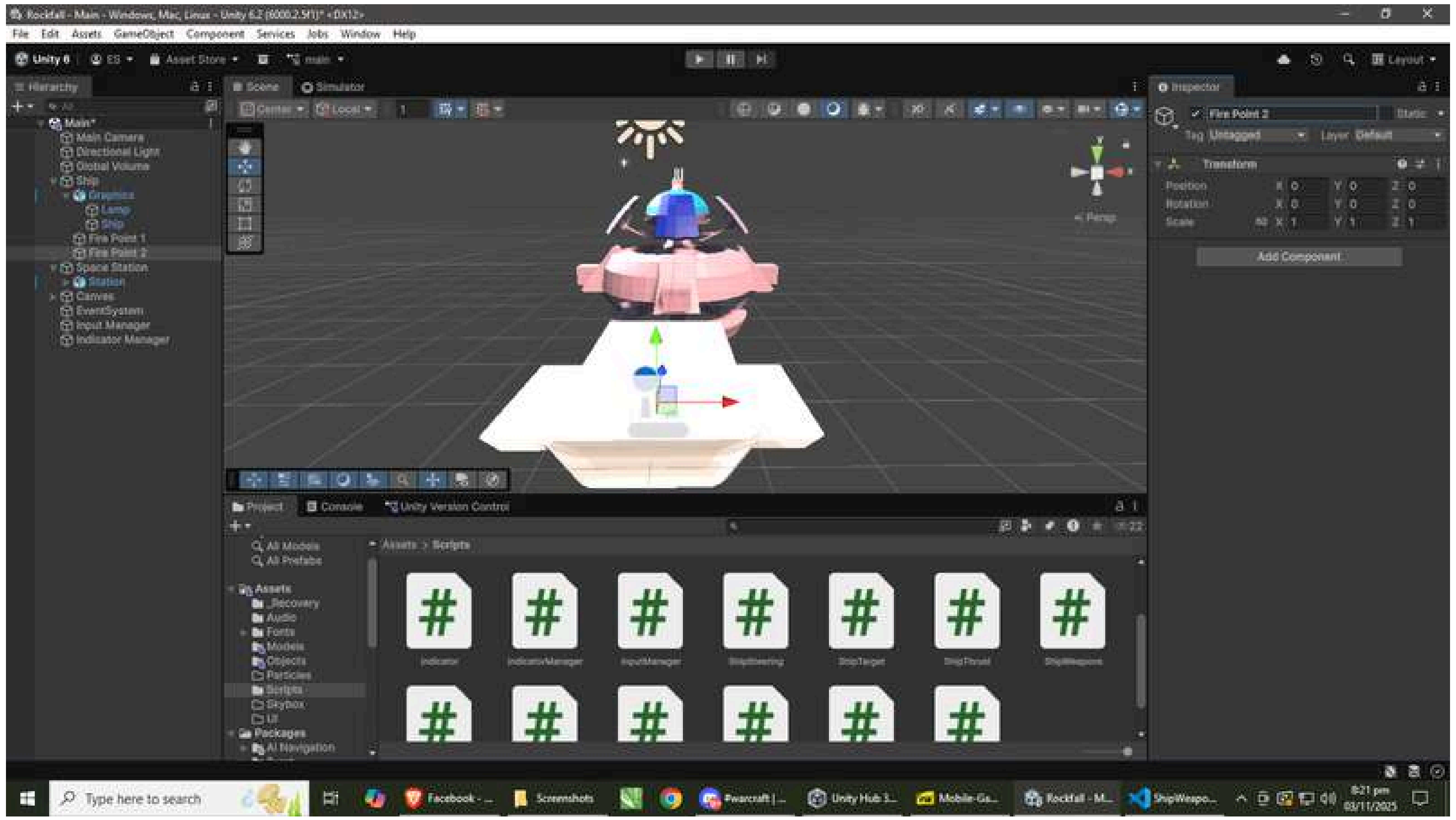
Add the following code to ShipWeapons.cs



```
using UnityEngine;
using System.Collections;

// BEGIN 3d_shipweapons
public class ShipWeapons : MonoBehaviour {

    // The prefab to use for each shot
    public GameObject shotPrefab;

    // BEGIN 3d_shipweapons_inputmanager
    public void Awake() {
        // When this object starts up, tell the input manager
        // to use me as the current weapon object
        InputManager.instance.SetWeapons(this);
    }

    // Called when the object is removed
    public void OnDestroy() {
        // Don't do this if we're not playing
        if (Application.isPlaying == true) {
            InputManager.instance.RemoveWeapons(this);
        }
    }
    // END 3d_shipweapons_inputmanager

    // The list of places where a shot can emerge from
    public Transform[] firePoints;

    // The index into firePoints that the next shot will fire from
    private int firePointIndex;

    // Called by InputManager.
    public void Fire() {

        // If we have no points to fire from, return
        if (firePoints.Length == 0)
            return;

        // Work out which point to fire from
```

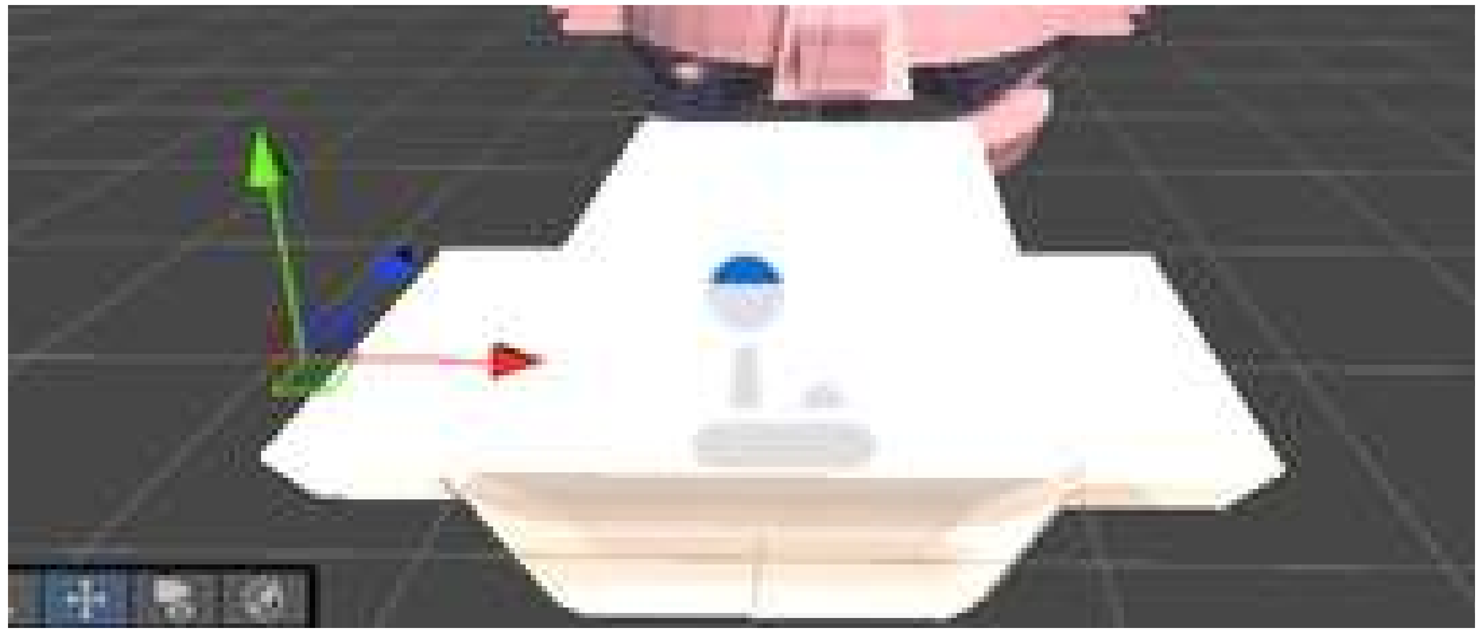
STEP 30

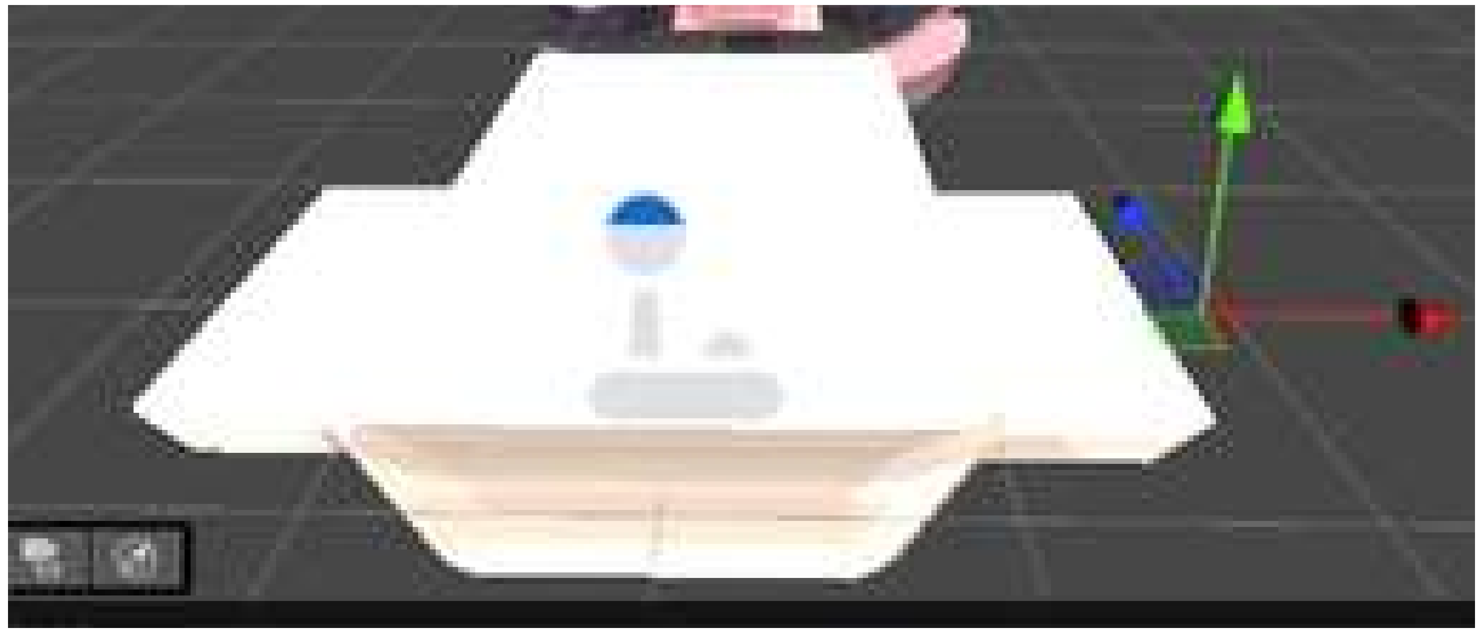
Create a new empty game object, and name it “Fire Point 1”. Make it a child of the Ship object, and then duplicate it by pressing Ctrl-D



STEP 31

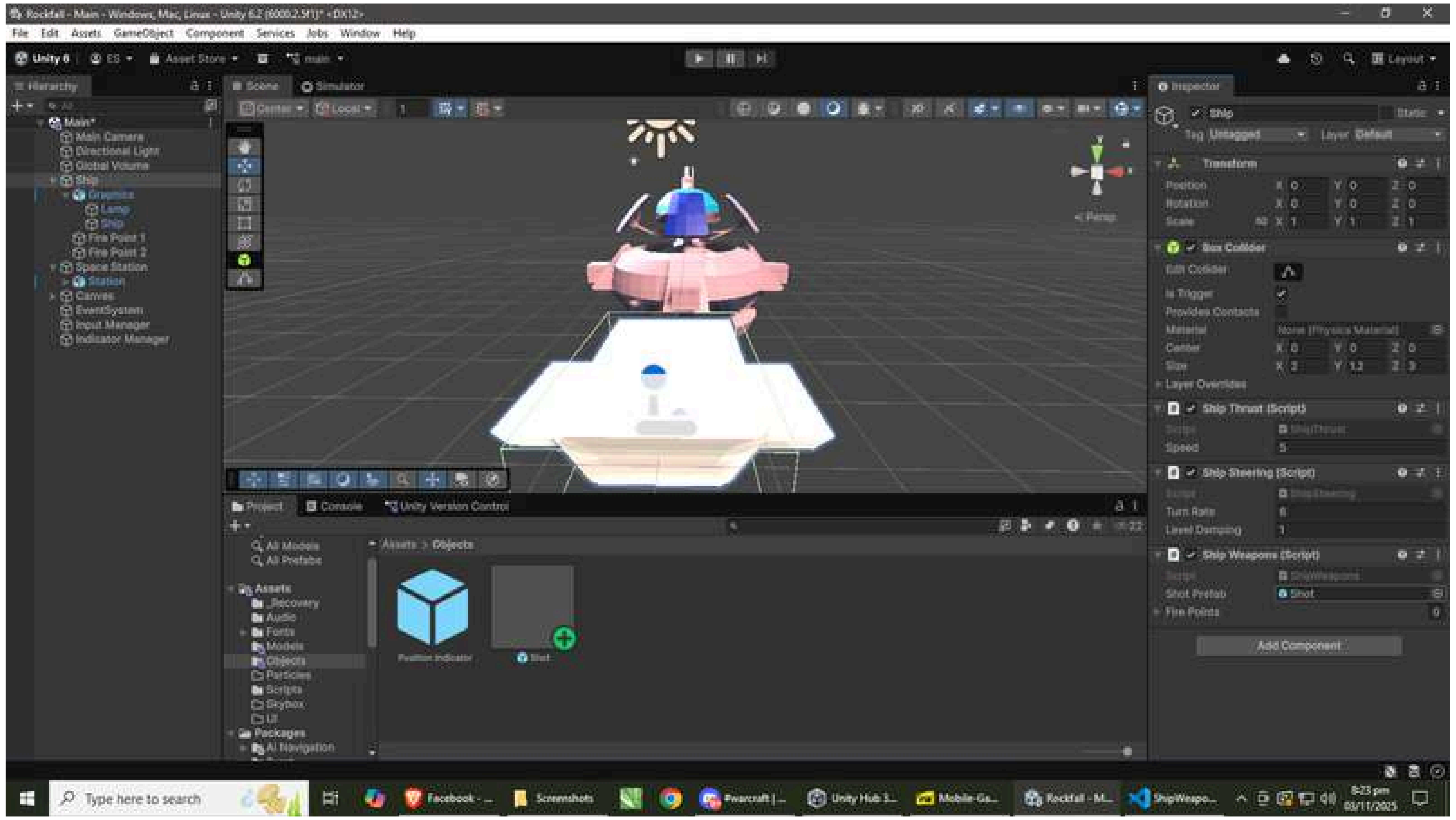
Set the position of Fire Point 1 to (-1.9, 0, 0). This will place it to the left of the ship. Set the position of Fire Point 2 to (1.9, 0, 0). This will place it to the right of the ship





STEP 32

Drag the Shot prefab that you created in the earlier section to the ShipWeapons's Shot Prefab slot.



Ship — Tag Untagged — Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Box Collider
Is Trigger ✓
Material None (Physics Material)
Center X 0 Y 0 Z 0
Size X 2 Y 1.2 Z 3

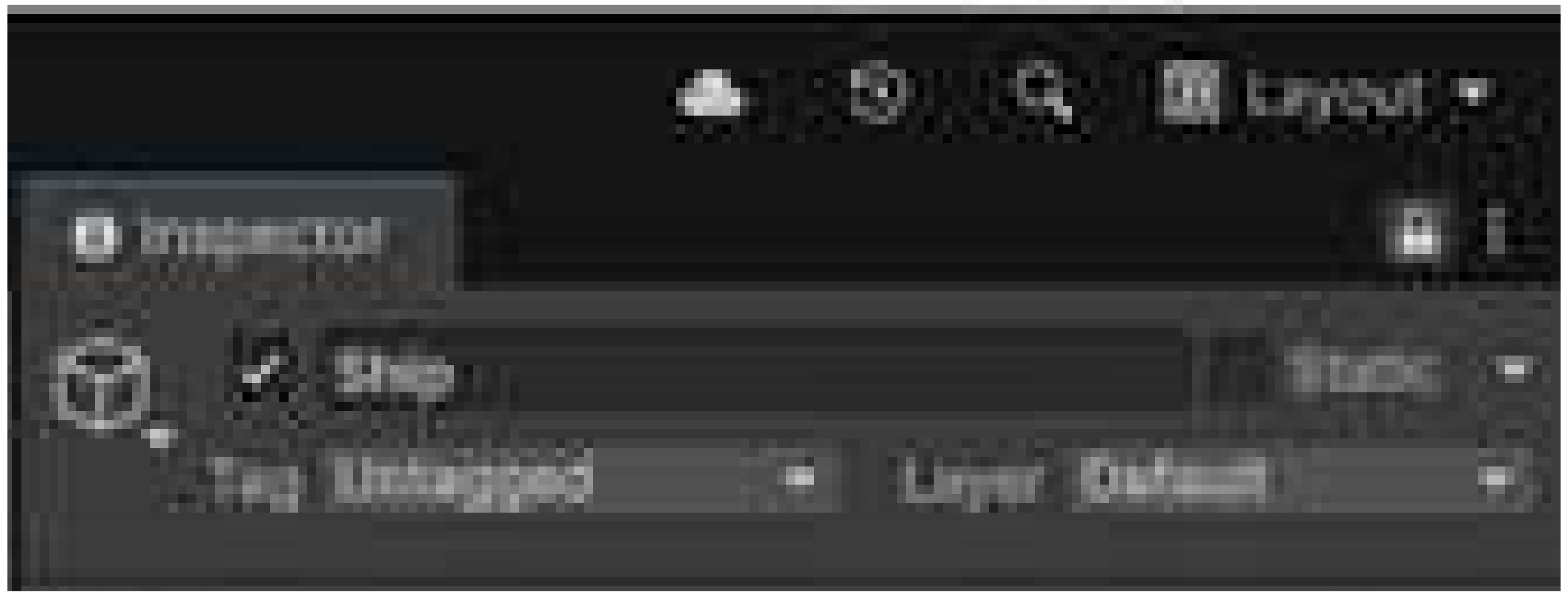
Ship Thrust (Script)
Script ShipThrust
Speed 5

Ship Steering (Script)
Script ShipSteering
Turn Rate 6
Level Damping 1

Ship Weapons (Script)
Script ShipWeapons
Shot Prefab Shot
Fire Points 0

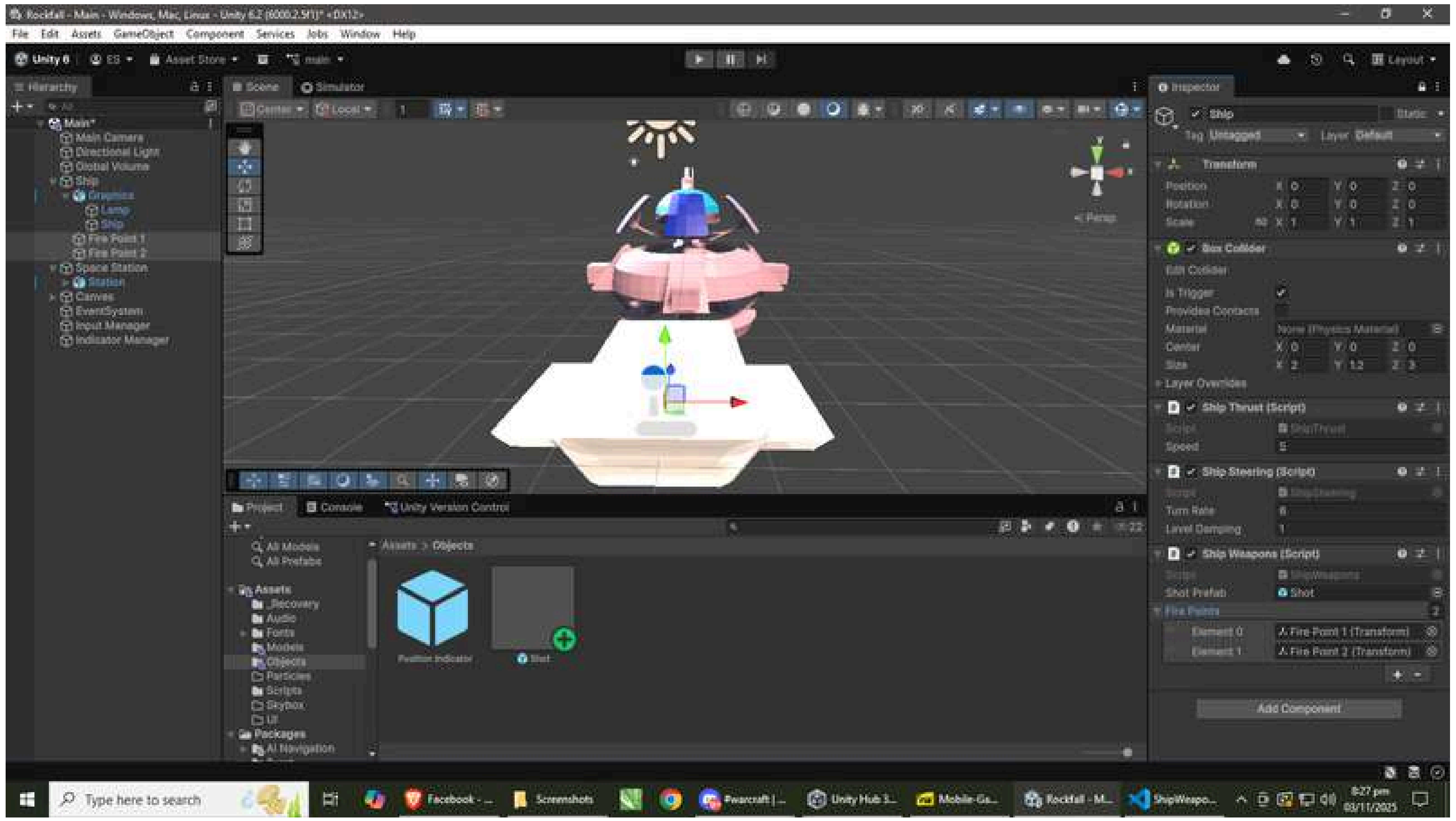
STEP 33

Select the Ship, and click the lock at the top right of the Inspector. This will lock the Inspector, and means that the object that the Inspector is showing won't change when you select another object.



STEP 34

Next, select both Fire Point objects in the Hierarchy by clicking Fire Point 1 and then holding the Ctrl key , and clicking Fire Point 2. Then, drag these two objects onto the ShipWeapons' Fire Points slot. Be sure to drag it onto the text “Fire Points” (and not any- thing below it), or it won’t work. After that unlock the inspector.



Ship — Tag Untagged — Layer Default

Transform: Position X 0 Y 0 Z 0; Rotation X 0 Y 0 Z 0; Scale X 1 Y 1 Z 1

Box Collider: Is Trigger ✓; Provides Contacts; Material None (Physics Material); Center X 0 Y 0 Z 0; Size X 2 Y 1.2 Z 3

Ship Thrust (Script): Script ShipThrust; Speed 5

Ship Steering (Script): Script ShipSteering; Turn Rate 6; Level Damping 1

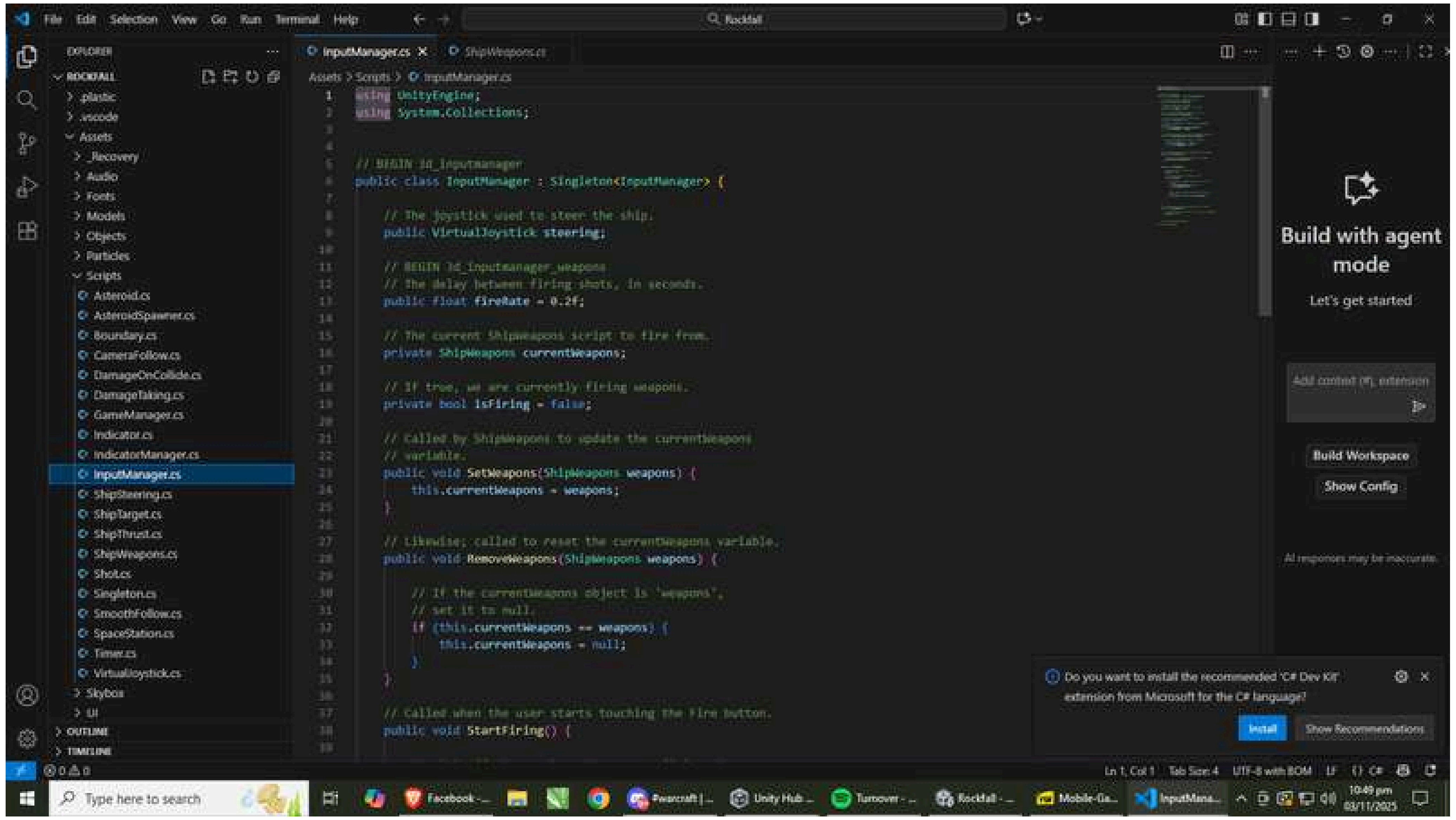
Ship Weapons (Script): Script ShipWeapons; Shot Prefab Shot

Fire Points 2
Element 0 Fire Point 1 (Transform)
Element 1 Fire Point 2 (Transform)

Add Component

STEP 35

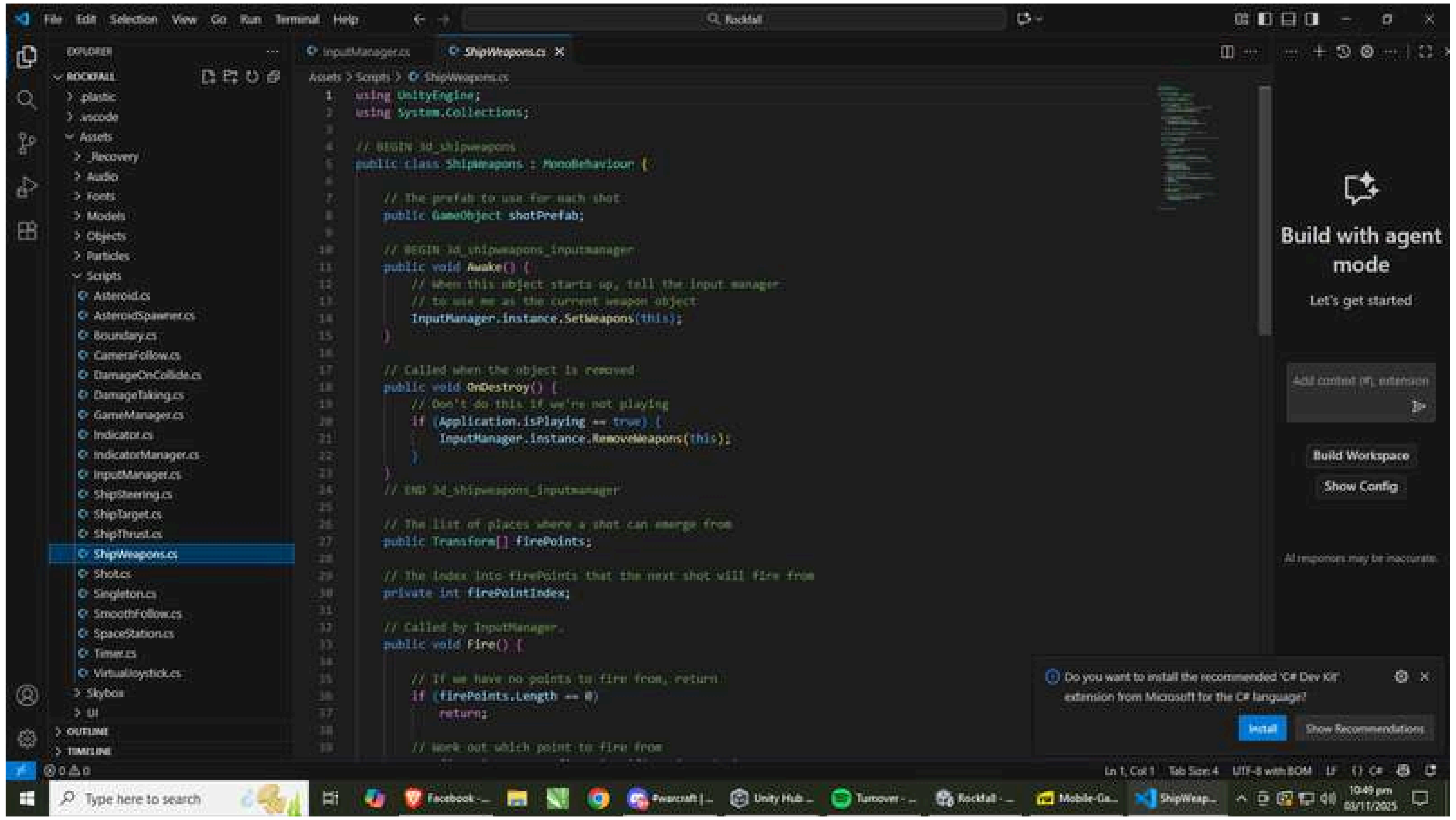
Add the ShipWeapons management code to InputManager, by adding the following properties and methods to the Input Manager class.



```
using UnityEngine;
using System.Collections;


// BEGIN 3d_inputmanager
public class InputManager : Singleton<InputManager> {

    // The joystick used to steer the ship.
    public VirtualJoystick steering;

    // BEGIN 3d_inputmanager_weapons
    // The delay between firing shots, in seconds.
    public float fireRate = 0.2f;

    // The current ShipWeapons script to fire from.
    private ShipWeapons currentWeapons;

    // If true, we are currently firing weapons.
    private bool isFiring = false;

    // Called by ShipWeapons to update the currentWeapons
    // variable.
    public void SetWeapons(ShipWeapons weapons) {
        this.currentWeapons = weapons;
    }

    // Likewise; called to reset the currentWeapons variable.
    public void RemoveWeapons(ShipWeapons weapons) {

        // If the currentWeapons object is 'weapons',
        // set it to null.
        if (this.currentWeapons == weapons) {
            this.currentWeapons = null;
        }
    }

    // Called when the user starts touching the fire button.
    public void StartFiring() {
```

STEP 36

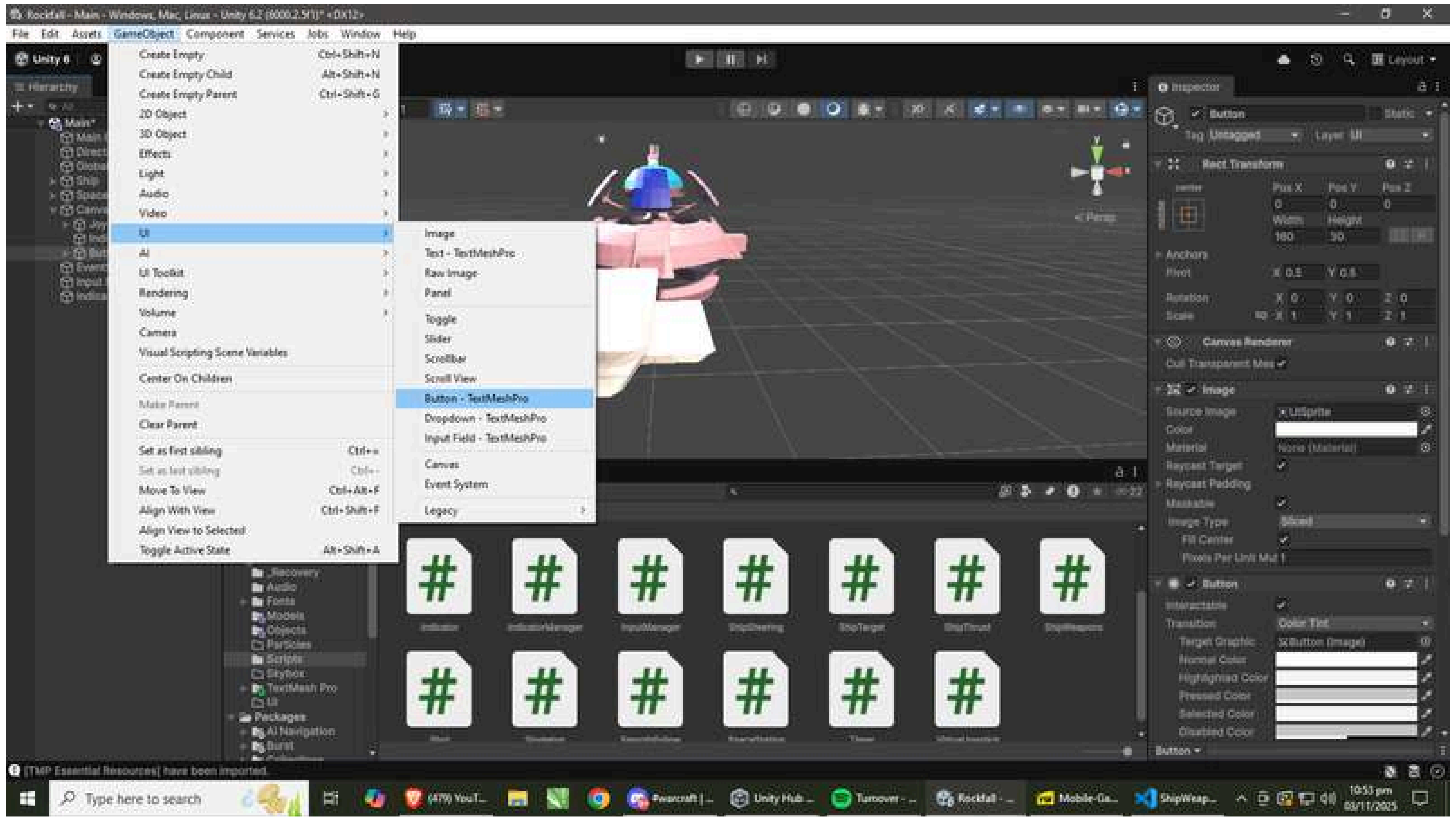
We then need to add the InputManager-communicating code to ShipWeapons by adding the following methods to the ShipWeapons class.



```
using UnityEngine;
using System.Collections;

// BEGIN 3d_shipweapons
public class ShipWeapons : MonoBehaviour {

    // The prefab to use for each shot
    public GameObject shotPrefab;

    // BEGIN 3d_shipweapons_inputmanager
    public void Awake() {
        // When this object starts up, tell the input manager
        // to use me as the current weapon object
        InputManager.instance.SetWeapons(this);
    }

    // Called when the object is removed
    public void OnDestroy() {
        // Don't do this if we're not playing
        if (Application.isPlaying == true) {
            InputManager.instance.RemoveWeapons(this);
        }
    }
    // END 3d_shipweapons_inputmanager

    // The list of places where a shot can emerge from
    public Transform[] firePoints;

    // The index into firePoints that the next shot will fire from
    private int firePointIndex;

    // Called by InputManager.
    public void Fire() {

        // If we have no points to fire from, return
        if (firePoints.Length == 0)
            return;

        // Work out which point to fire from
```

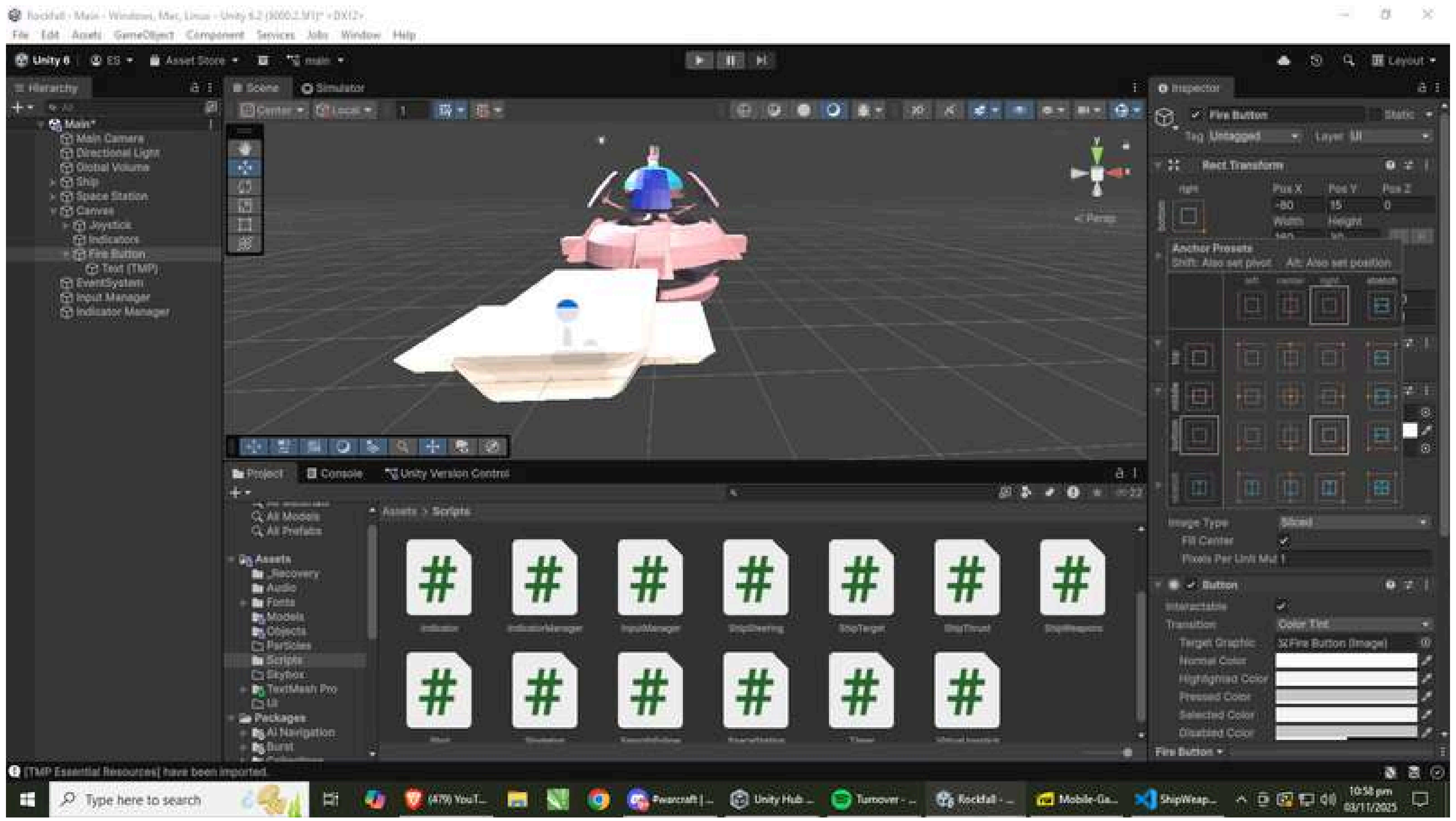
STEP 37

Create a new button by opening the GameObject menu, and choos- ing UI → Button. Name the new button "Fire Button".



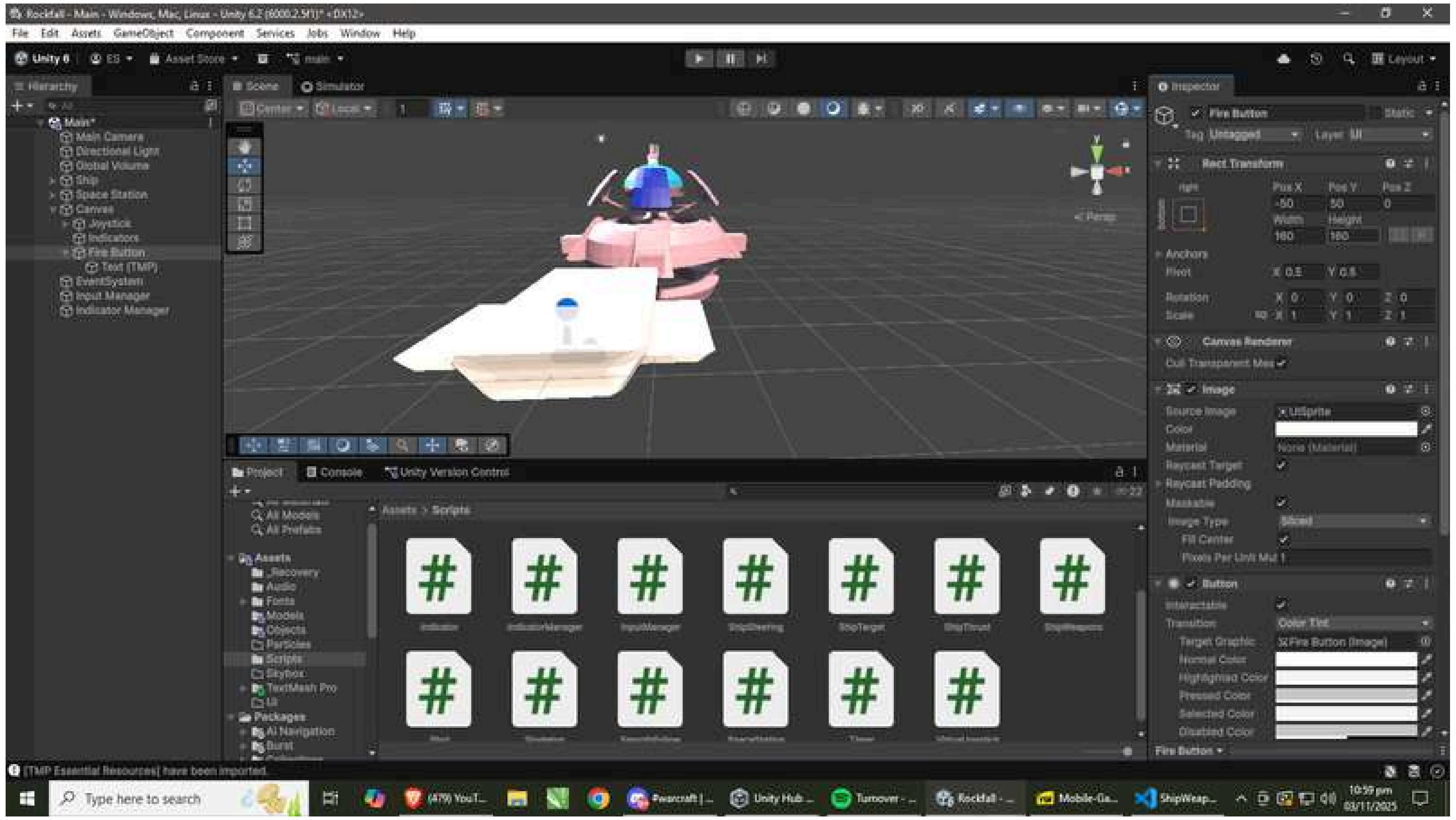
STEP 38

Set both the anchors and the pivot of the button to Bottom Right by clicking on the Anchor button at the top-left of the Inspector, hold- ing the Alt key , and clicking on the Bottom Right option.



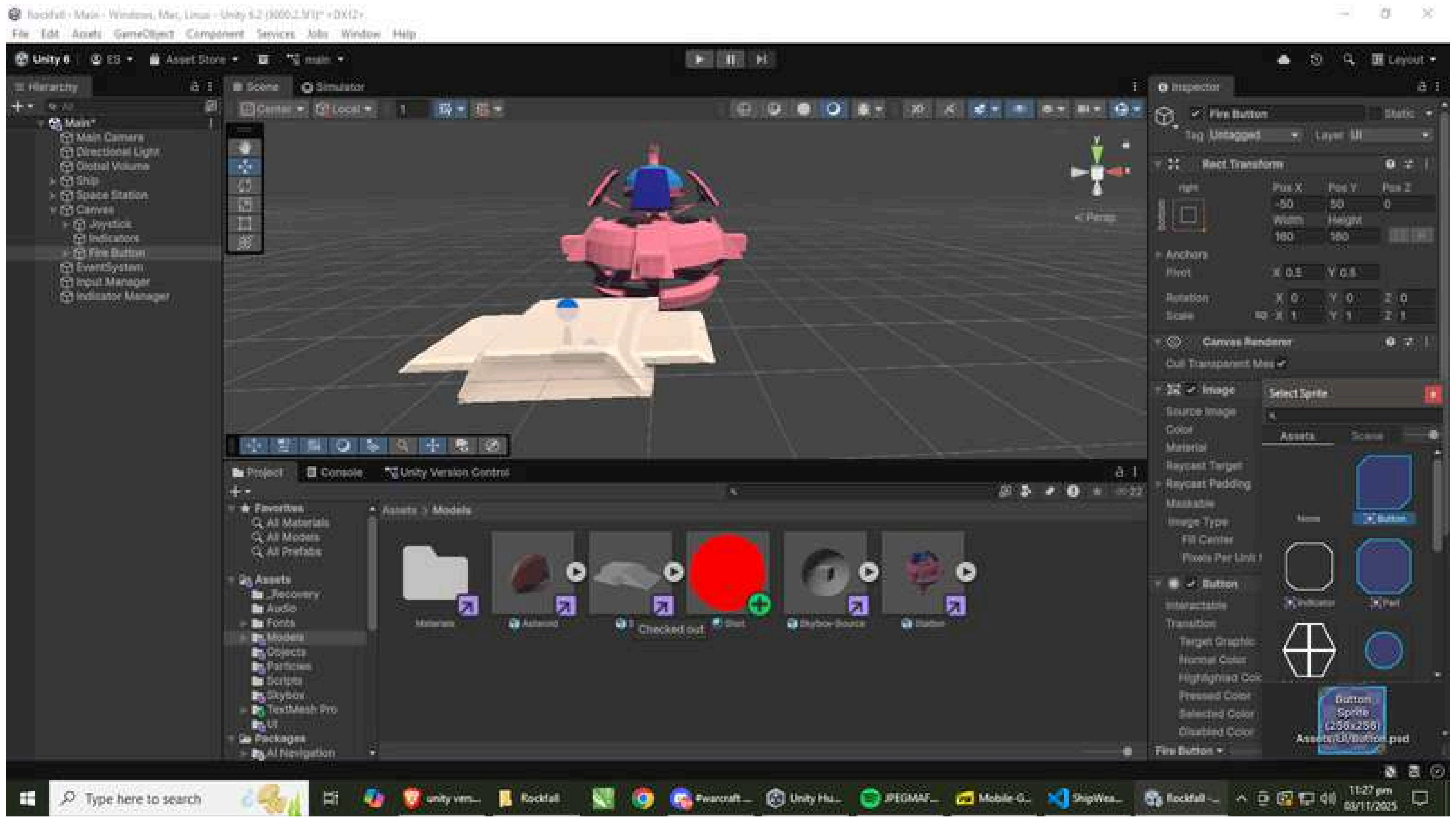
STEP 39

Next, set the position of the button to (-50, 50, 0). This will place the button at the bottom-right of the canvas. Set both the width and height of the button to 160.


Fire Button
Rect Transform
Pos X -50 Pos Y 50 Pos Z 0
Width 160 Height 160
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

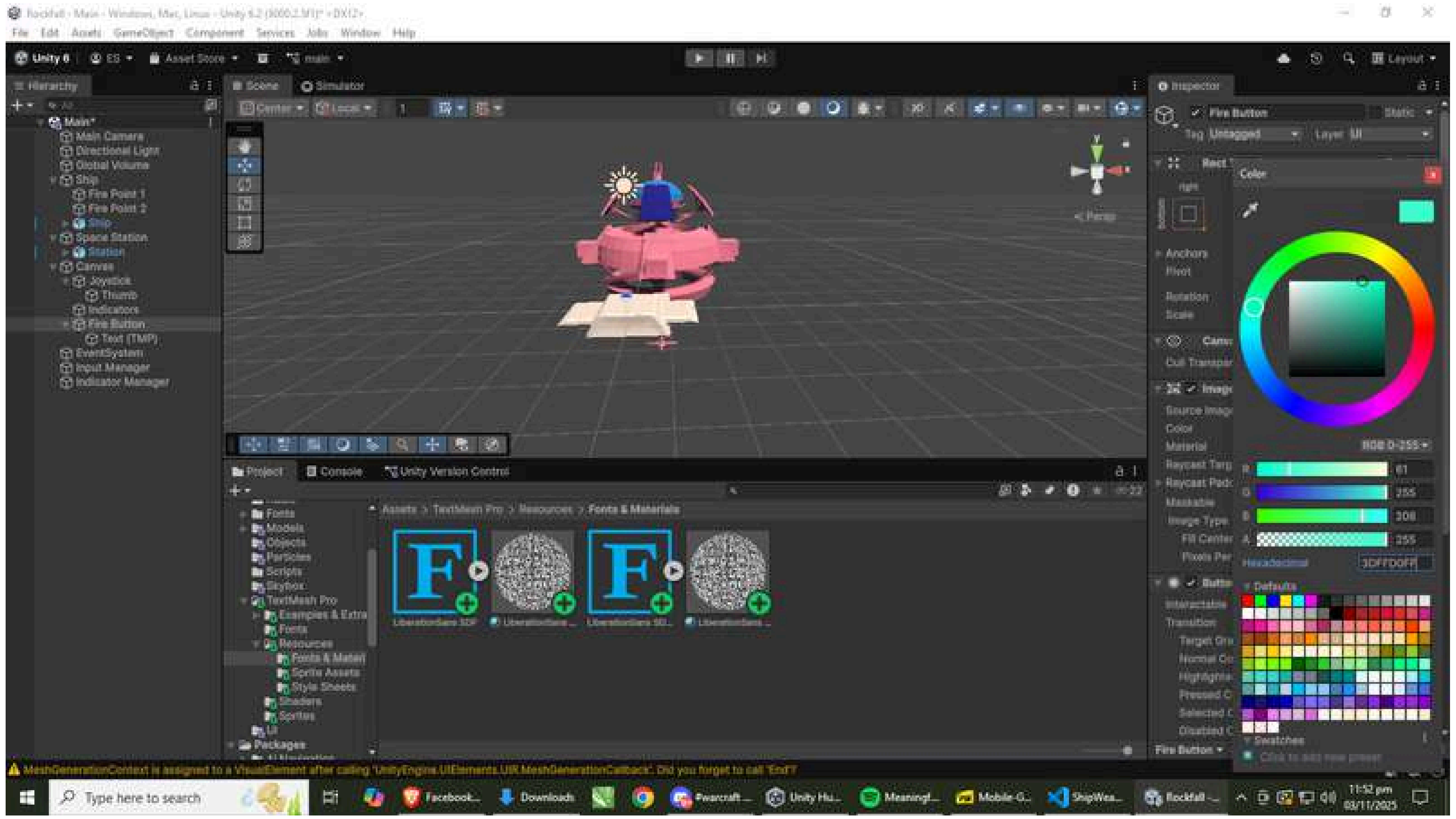
STEP 40

Set the Source Image of the button’s Image component to the Button sprite. Set the Image Type to Sliced.



STEP 41

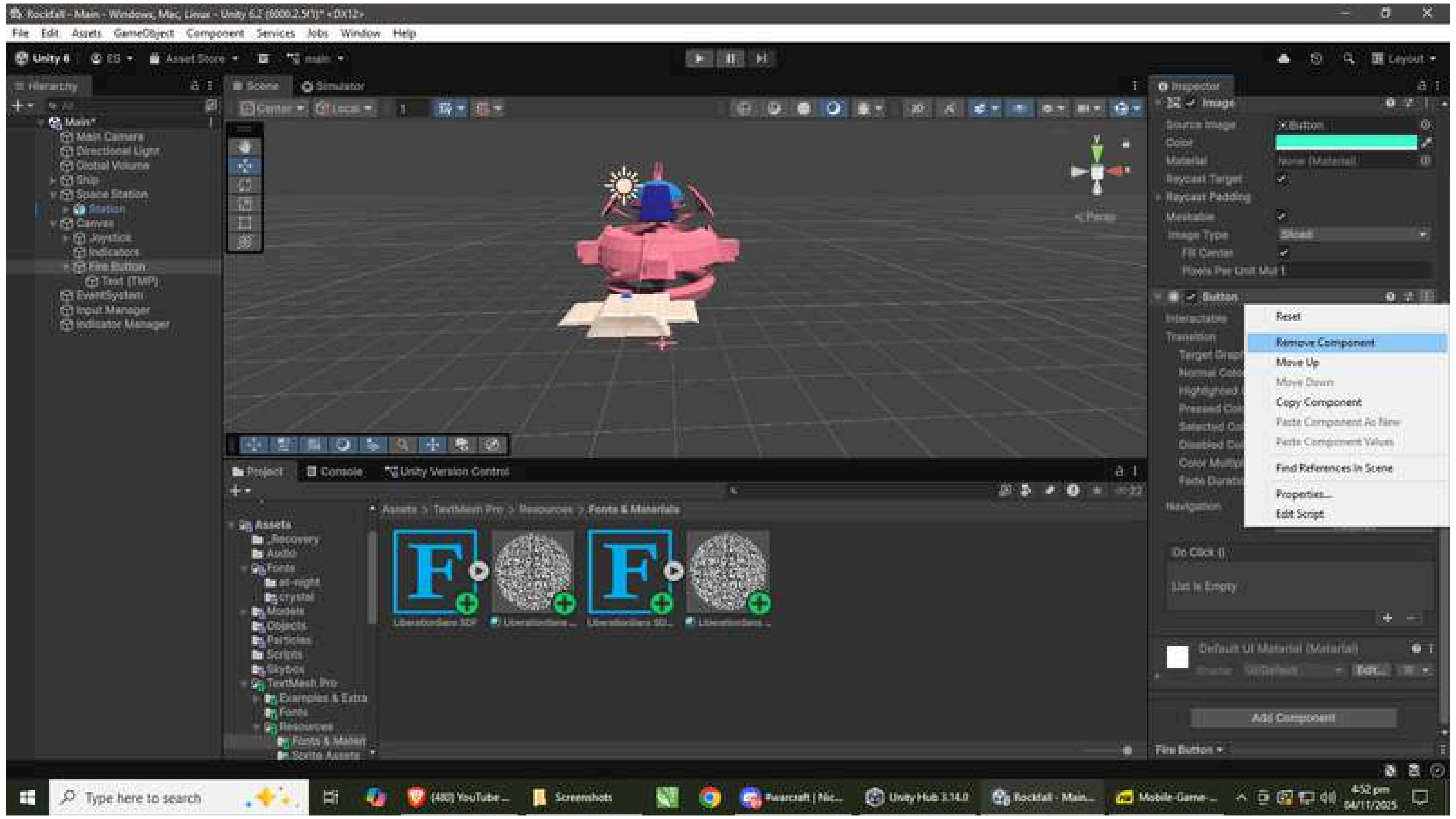
Set the color of the Fire button to a light cyan by clicking on the Color field, and in the Hex Color field, enter 3DFFD0FF.



R 61
G 255
B 208
A 255
Hexadecimal 3DFFD0FF

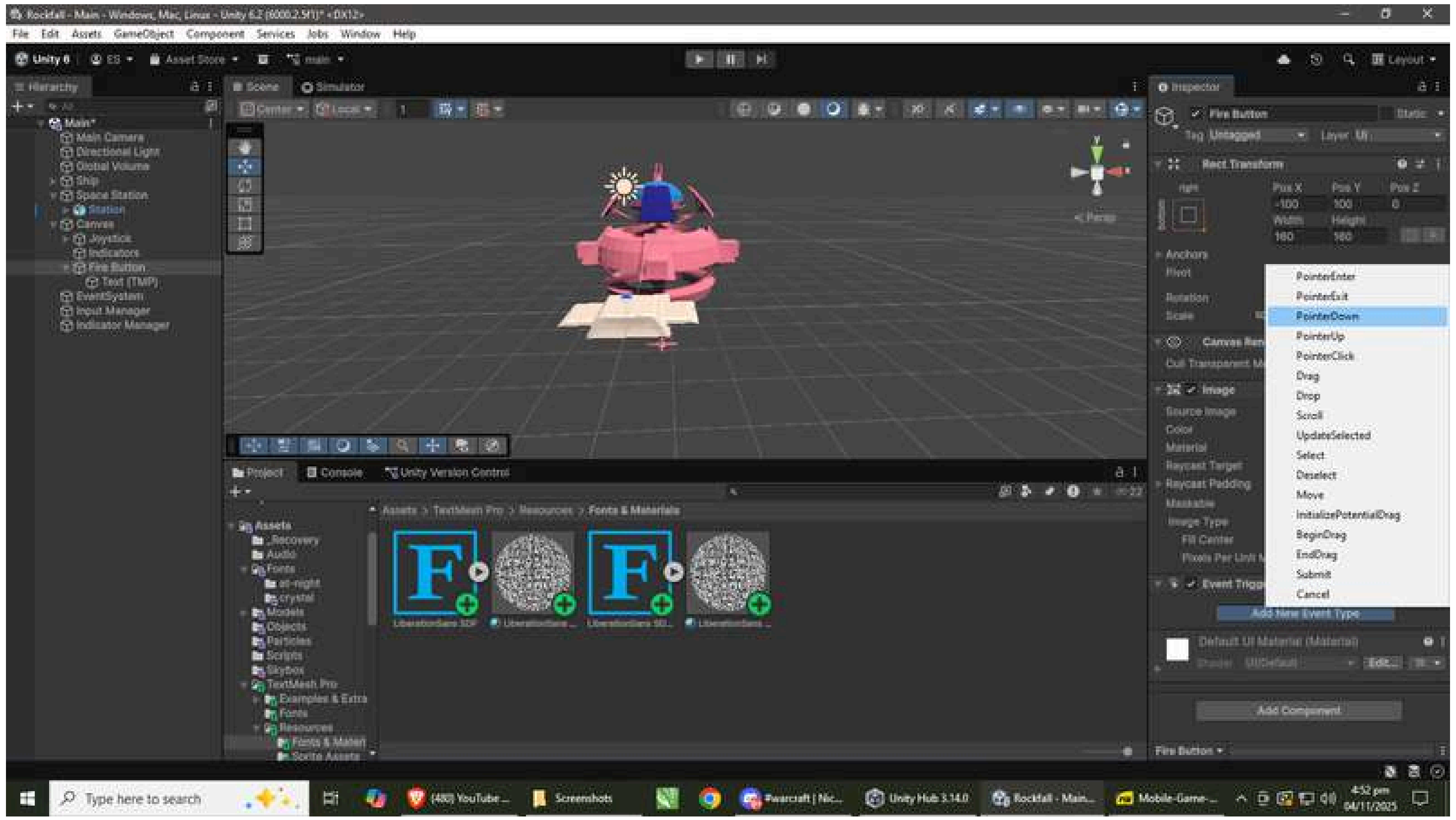
●●● STEP 42 ✕

Select the Fire Button object, and click on the settings icon at the top right of the Button compo- nent. Click Remove Component.



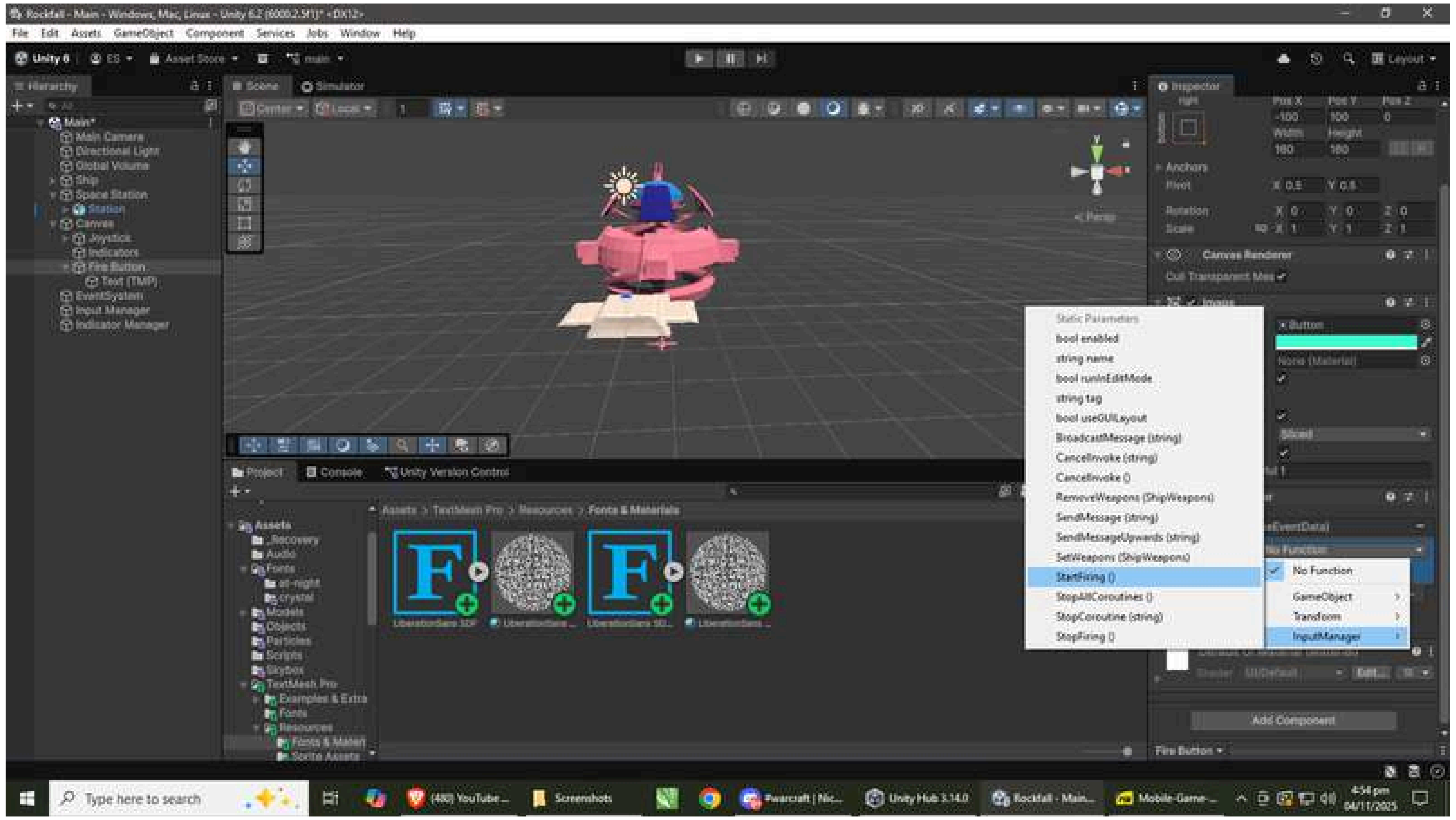
STEP 43

Add a new Event Trigger component, and then click Add Event Type. Choose “PointerDown” from the menu that appears.



STEP 44

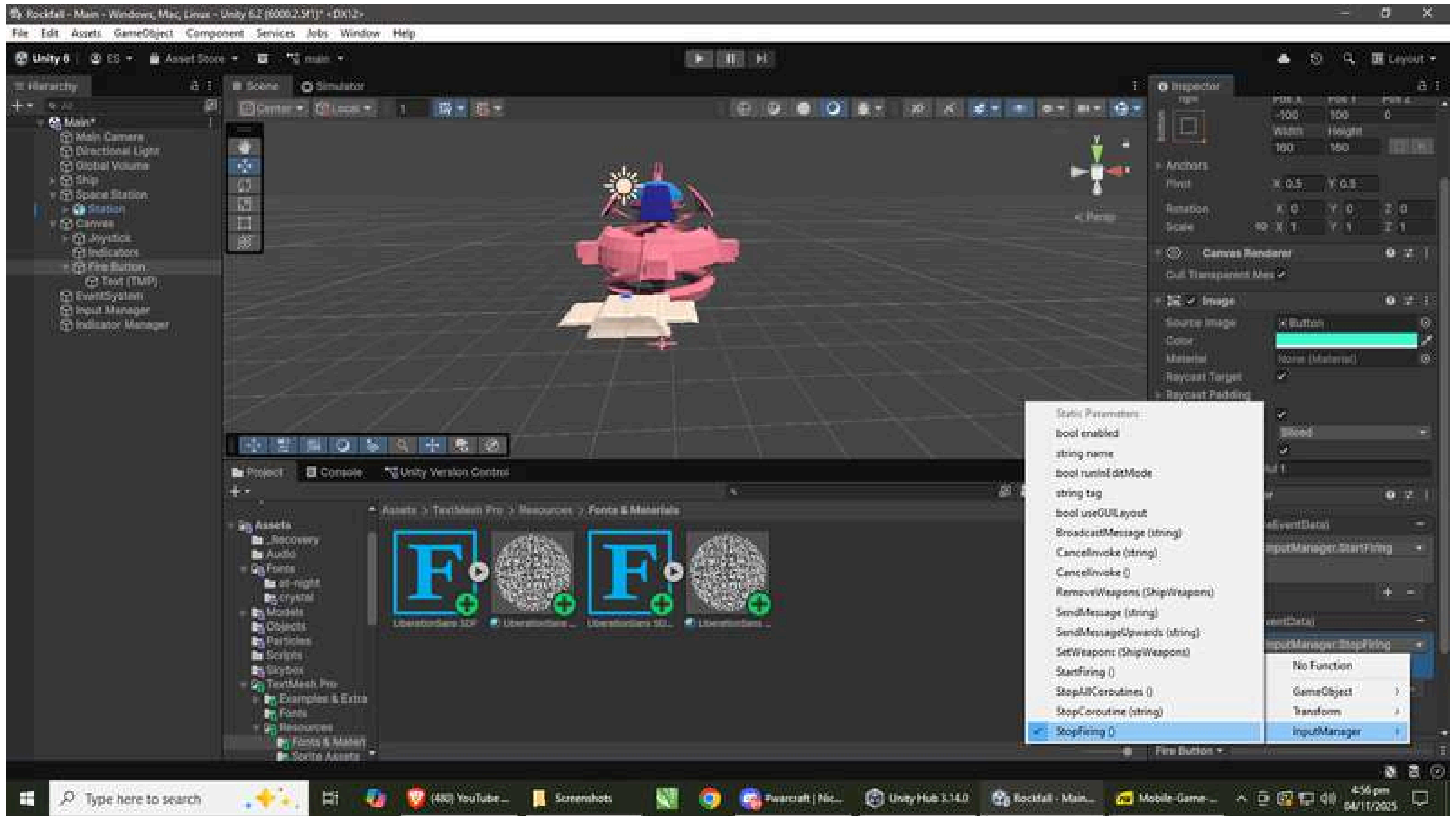
Click the + button at the bottom of the PointerDown list, and a new item will appear in the list. Drag and drop the Input Manager object from the Hierarchy panel into the slot. Next, change the method from “No Function” to “InputManager→StartFiring”.



●●●

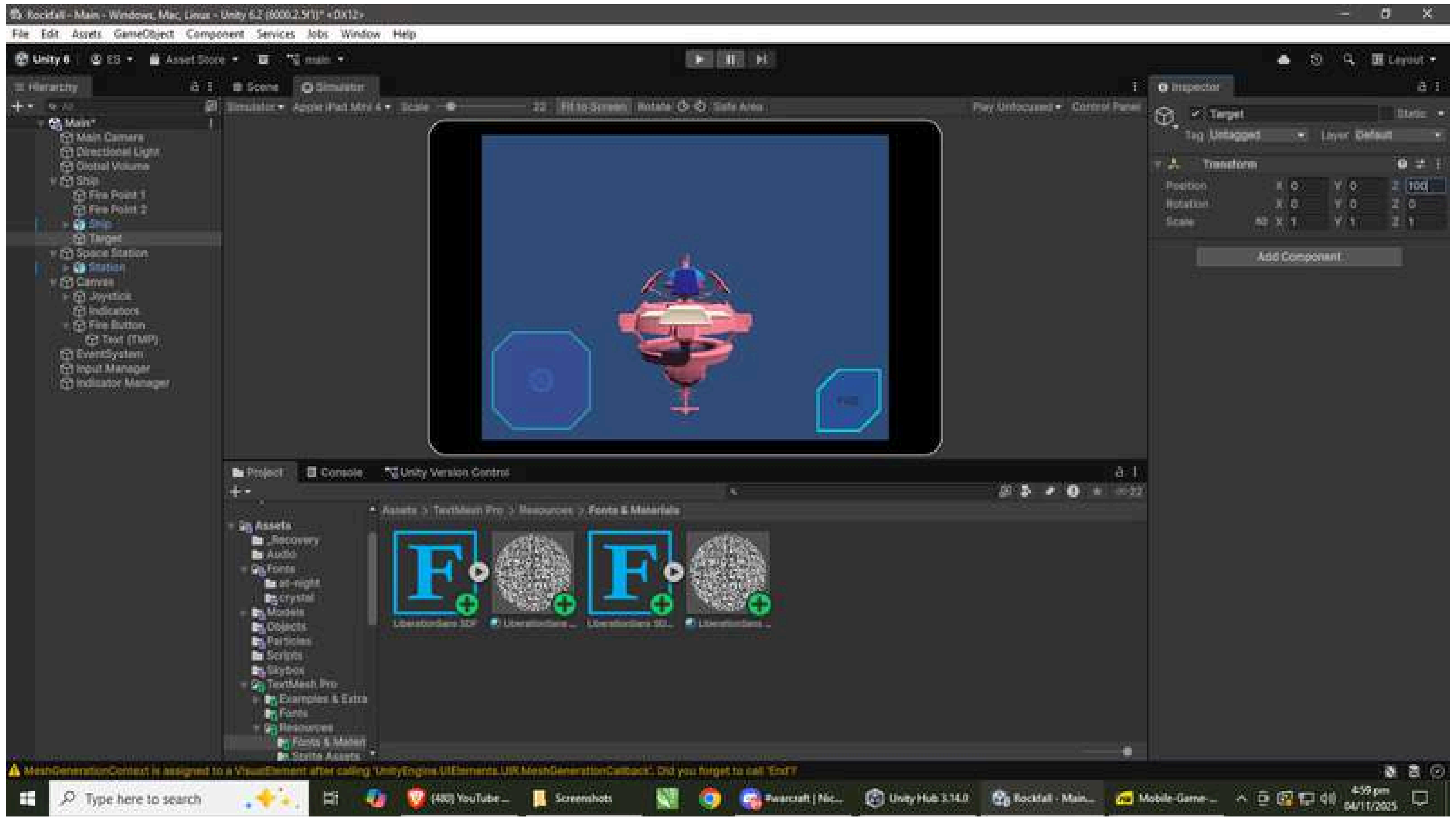
STEP 45

Next, you need to add an event for when the finger lifts up from the screen. Click the Add Event Type again, and choose “PointerUp”. Configure this event in the same way as the PointerDown, but make the method called on the InputManager be “StopFiring”



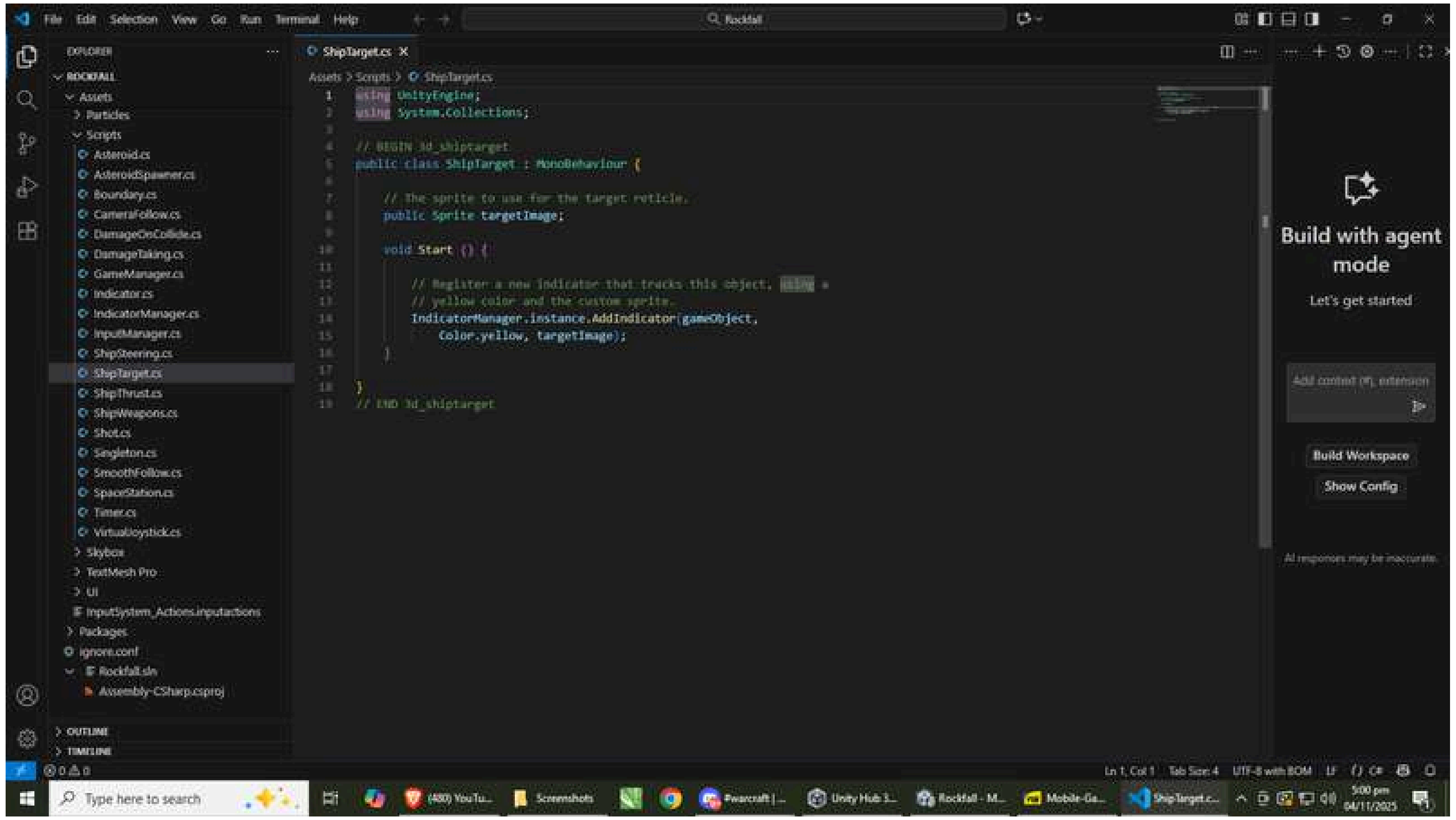
STEP 46

Name this object “Target”, and make it a child of the Ship. Position the Target. Set the position of the Target object to (0,0,100). This will place the target some distance away from the ship.



STEP 47

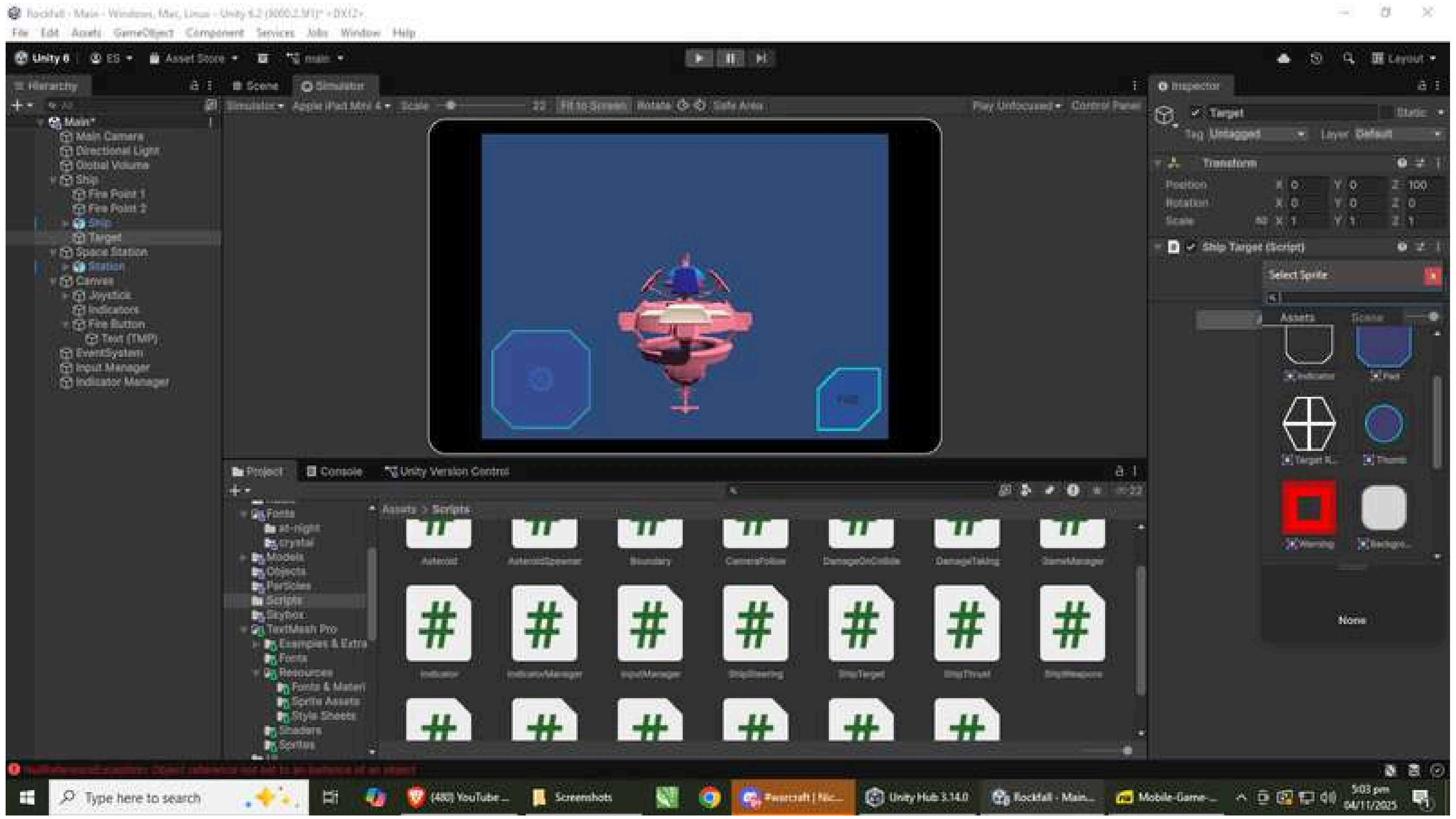
Add a new C# script to the Target object named ShipTarget.cs, and add the following code to it.



```
using UnityEngine;
using System.Collections;

// BEGIN 3d_shiptarget
public class ShipTarget : MonoBehaviour {

    // The sprite to use for the target reticle.
    public Sprite targetImage;

    void Start () {

        // Register a new indicator that tracks this object, using a
        // yellow color and the custom sprite.
        IndicatorManager.instance.AddIndicator(gameObject,
            Color.yellow, targetImage);
    }

}
// END 3d_shiptarget
```

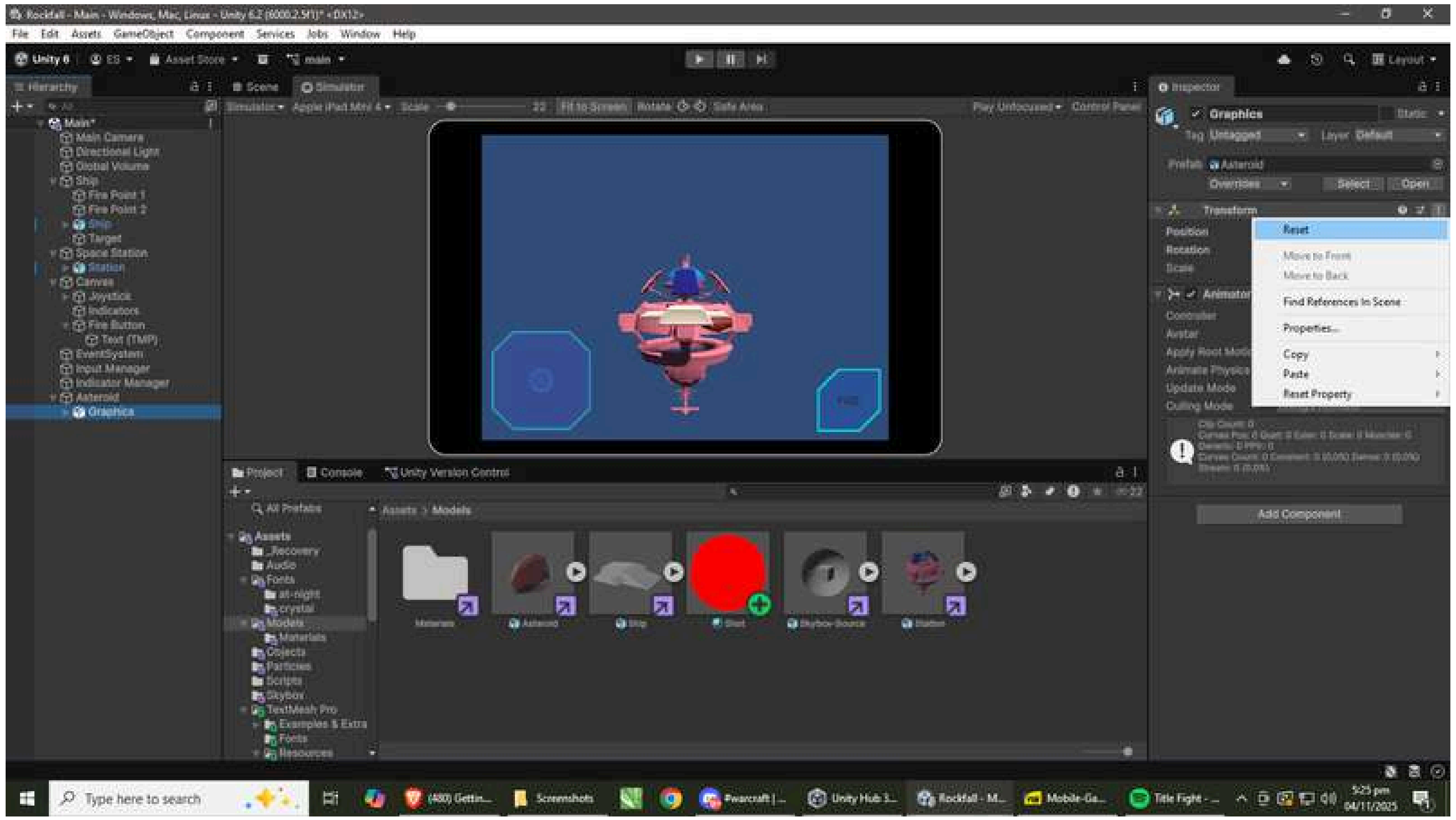
STEP 48

Drag the “Target Reticle” sprite into the Target Image slot of the ShipTarget script.



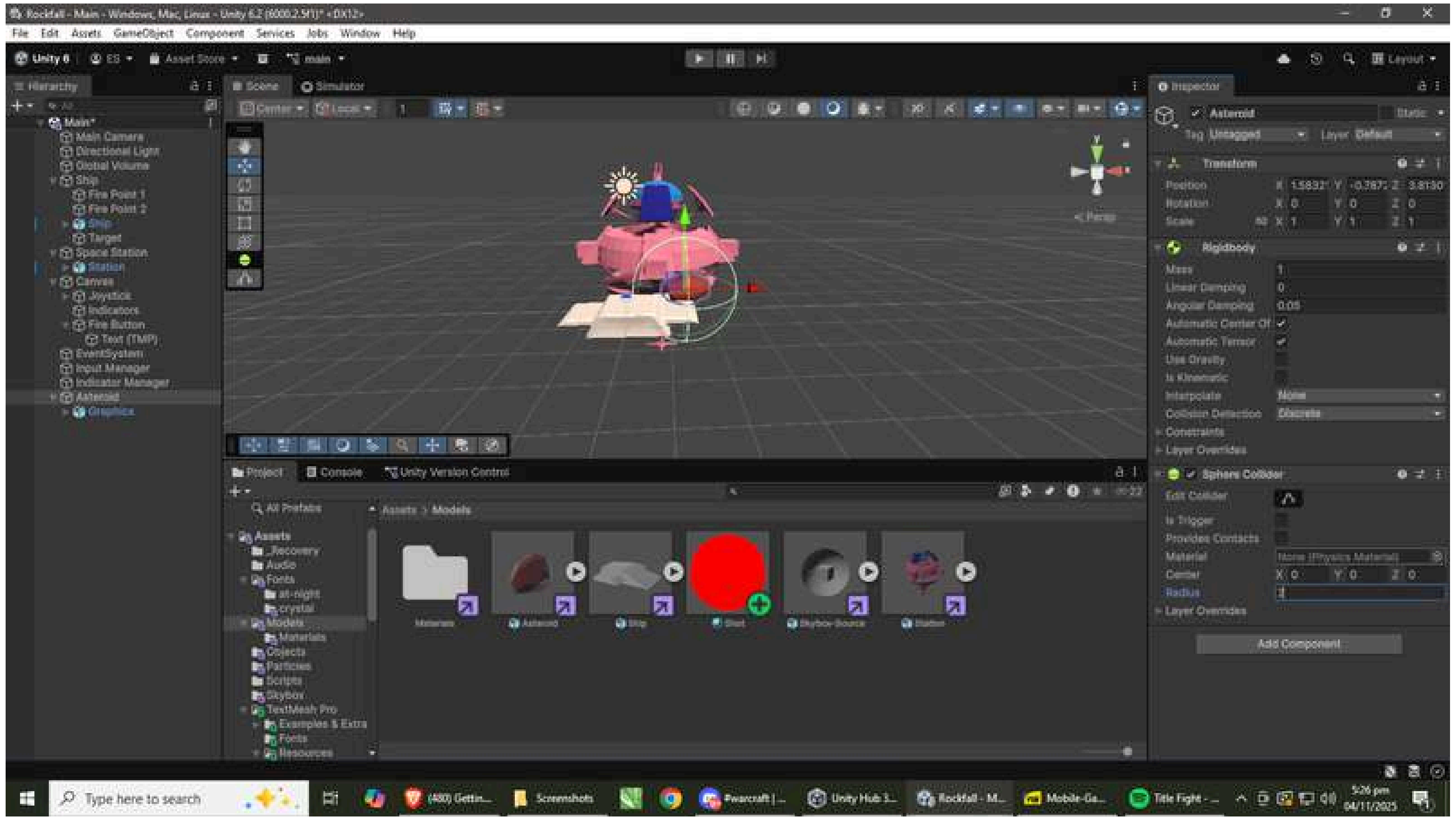
STEP 49

Make a new game object named "Asteroid". Add the asteroid model to it. Locate the Asteroid model in the Models folder. Drag it onto the Asteroid object you just created, and rename the new child object "Graphics". Reset the Position of the Graphics object's Transform component, so that it's positioned at (0,0,0).



STEP 50

Add a rigidbody and sphere collider to the Asteroid object. Don't add it to the Graphics object. Once they're added, turn gravity off on the rigidbody, and make the radius of the sphere collider be 2.

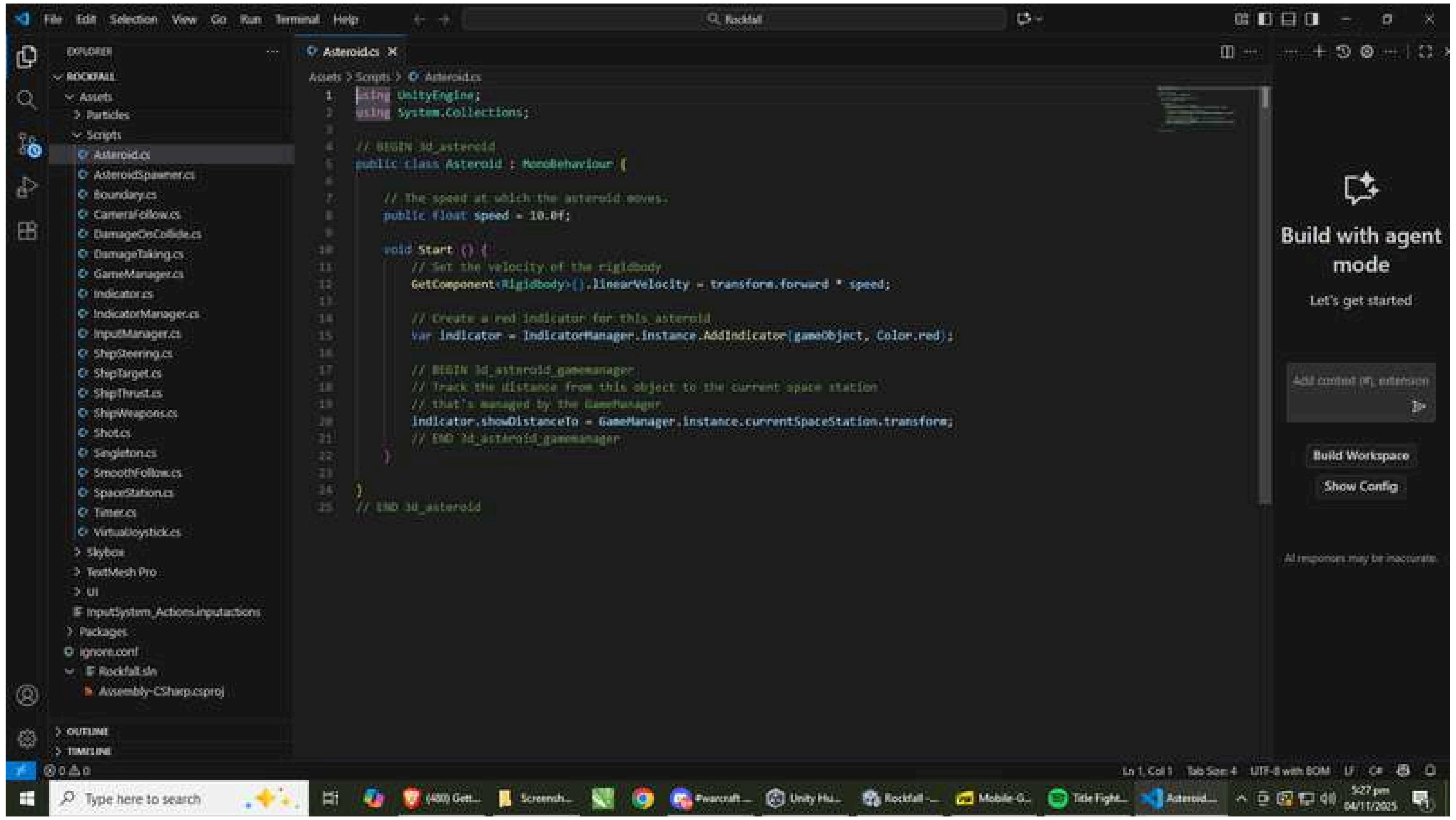


Rockfall - Main - Windows, Mac, Linux - Unity 6.2 (6000.2.5f1)* <DX12>
File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
Scene
Simulator
Project
Console
Unity Version Control
Inspector

```
Asteroid                         Static
Tag Untagged        Layer Default
Transform
  Position   X 1.5832   Y -0.787   Z 3.8130
  Rotation   X 0        Y 0        Z 0
  Scale      X 1        Y 1        Z 1
Rigidbody
  Mass                     1
  Linear Damping           0
  Angular Damping          0.05
  Automatic Center Of      ✓
  Automatic Tensor         ✓
  Use Gravity
  Is Kinematic
  Interpolate              None
  Collision Detection      Discrete
  Constraints
  Layer Overrides
Sphere Collider
  Edit Collider
  Is Trigger
  Provides Contacts
  Material                 None (Physics Material)
  Center     X 0   Y 0   Z 0
  Radius                   2
  Layer Overrides
Add Component
```

STEP 51

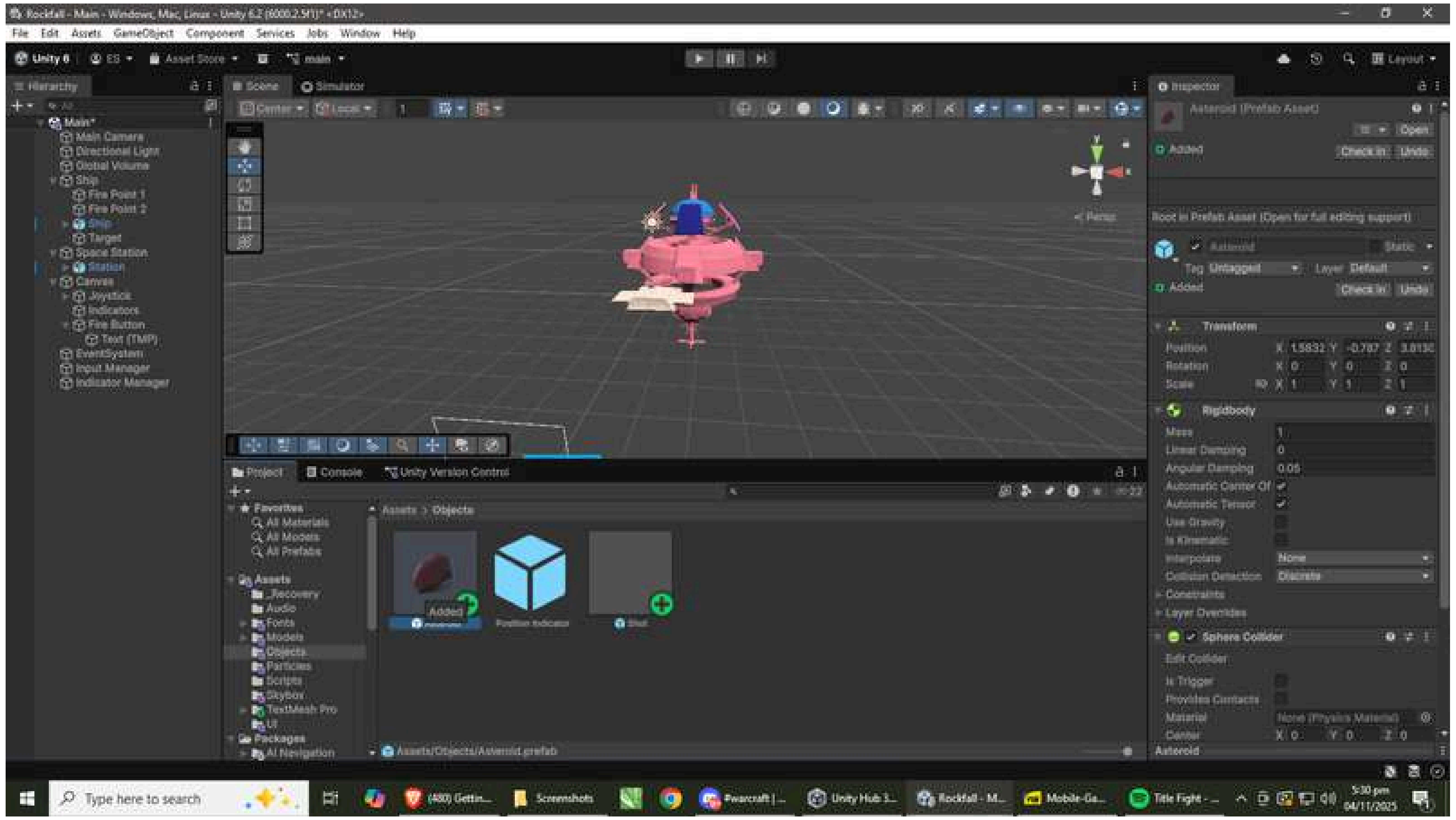
Add a new C# script to the Asteroid game object, called Asteroid.cs, and add the following code to it.



```
using UnityEngine;
using System.Collections;

// BEGIN 3d_asteroid
public class Asteroid : MonoBehaviour {

    // The speed at which the asteroid moves.
    public float speed = 10.0f;

    void Start () {
        // Set the velocity of the rigidbody
        GetComponent<Rigidbody>().linearVelocity = transform.forward * speed;

        // Create a red indicator for this asteroid
        var indicator = IndicatorManager.instance.AddIndicator(gameObject, Color.red);

        // BEGIN 3d_asteroid_gamemanager
        // Track the distance from this object to the current space station
        // that's managed by the GameManager
        indicator.showDistanceTo = GameManager.instance.currentSpaceStation.transform;
        // END 3d_asteroid_gamemanager
    }

}
// END 3d_asteroid
```

STEP 52

Drag the Asteroid object from the Hierarchy panel into the Project panel. This will create a prefab from the object. Next, delete the Asteroid from the scene.

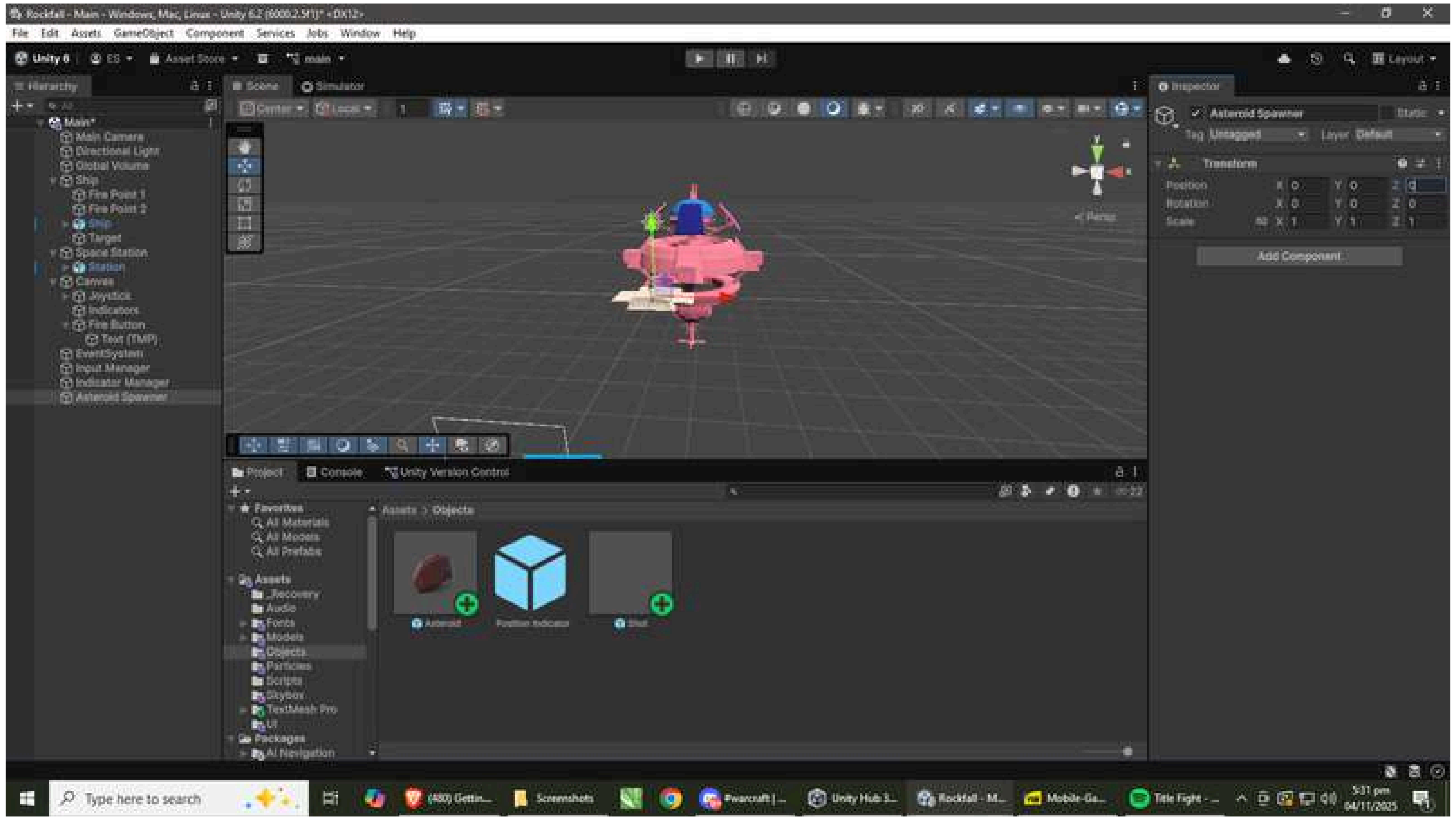


●●●

STEP 53

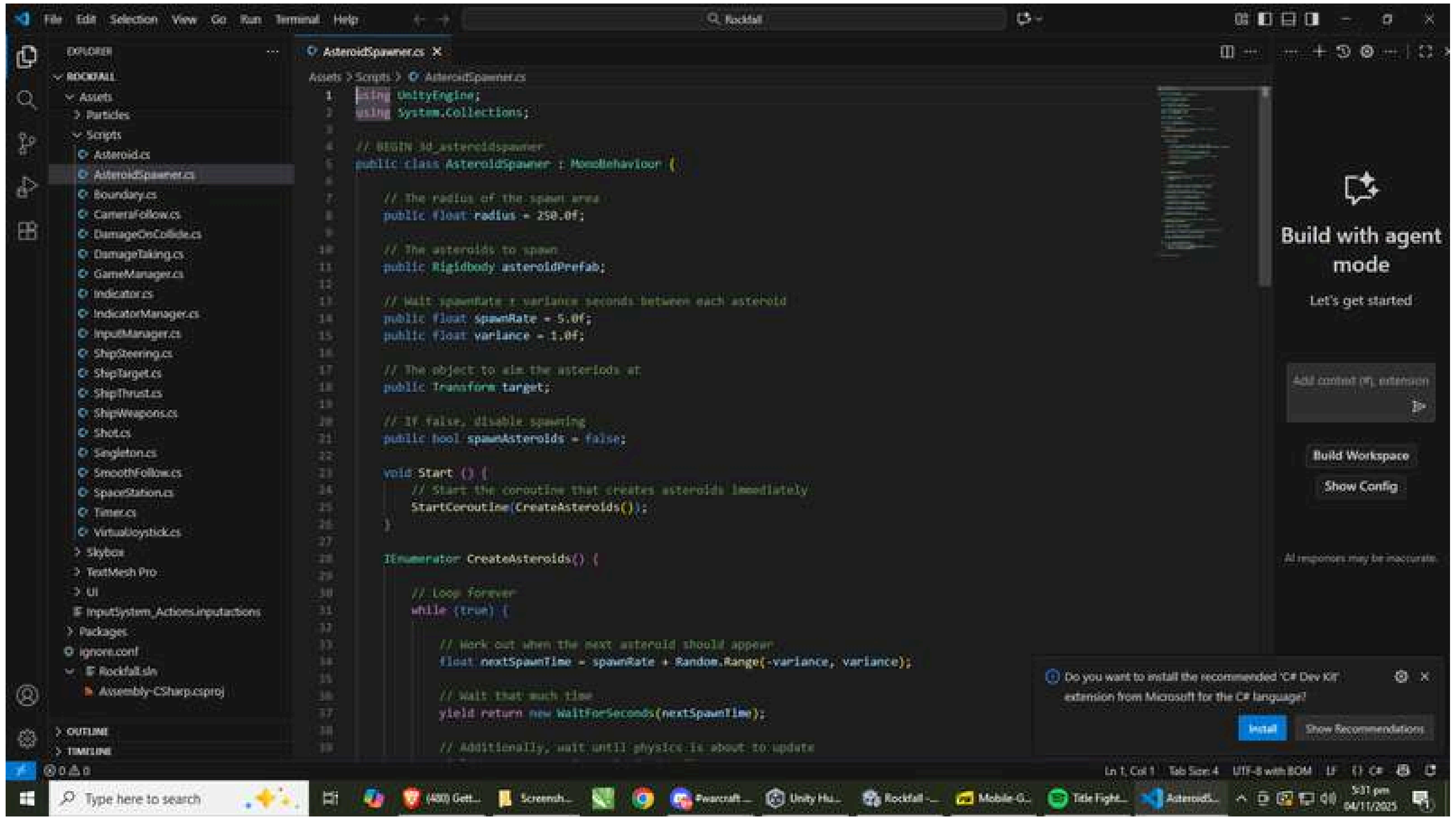
✕

Make a new empty game object, and name it “Asteroid Spawner”. Set its position to (0,0,0).



STEP 54

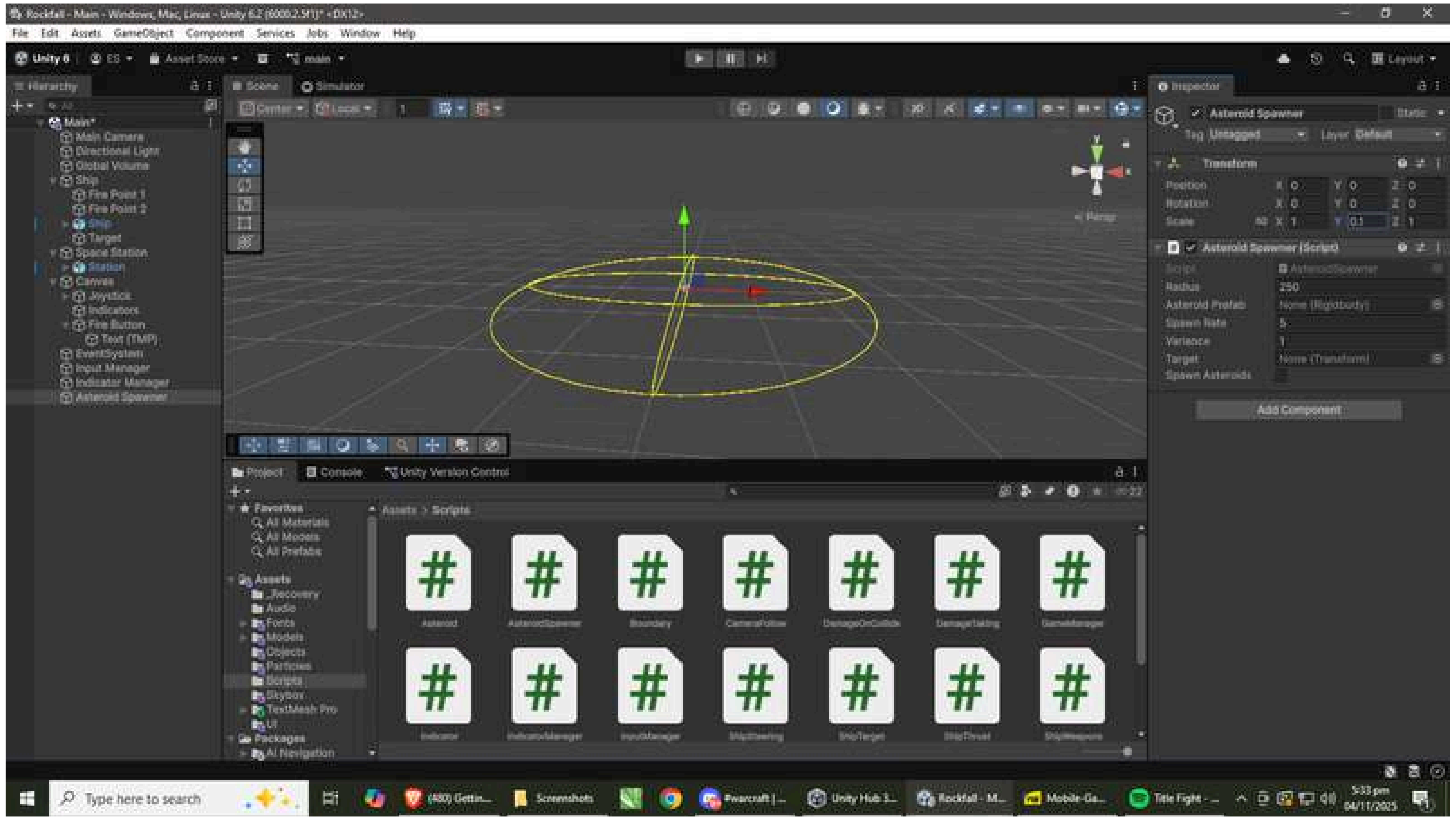
Add a new C# script called AsteroidSpawner.cs, and add the following code to it



```
using UnityEngine;
using System.Collections;

// BEGIN 3d_asteroidspawner
public class AsteroidSpawner : MonoBehaviour {

    // The radius of the spawn area
    public float radius = 250.0f;

    // The asteroids to spawn
    public Rigidbody asteroidPrefab;

    // Wait spawnRate ± variance seconds between each asteroid
    public float spawnRate = 5.0f;
    public float variance = 1.0f;

    // The object to aim the asteriods at
    public Transform target;

    // If false, disable spawning
    public bool spawnAsteroids = false;

    void Start () {
        // Start the coroutine that creates asteroids immediately
        StartCoroutine(CreateAsteroids());
    }

    IEnumerator CreateAsteroids() {

        // Loop forever
        while (true) {

            // Work out when the next asteroid should appear
            float nextSpawnTime = spawnRate + Random.Range(-variance, variance);

            // Wait that much time
            yield return new WaitForSeconds(nextSpawnTime);

            // Additionally, wait until physics is about to update
```

STEP 55

Set the Asteroid Spawner's Scale to (1,0.1,1). Doing this will make the asteroids mostly appear in a circle around their target, rather than in a sphere.



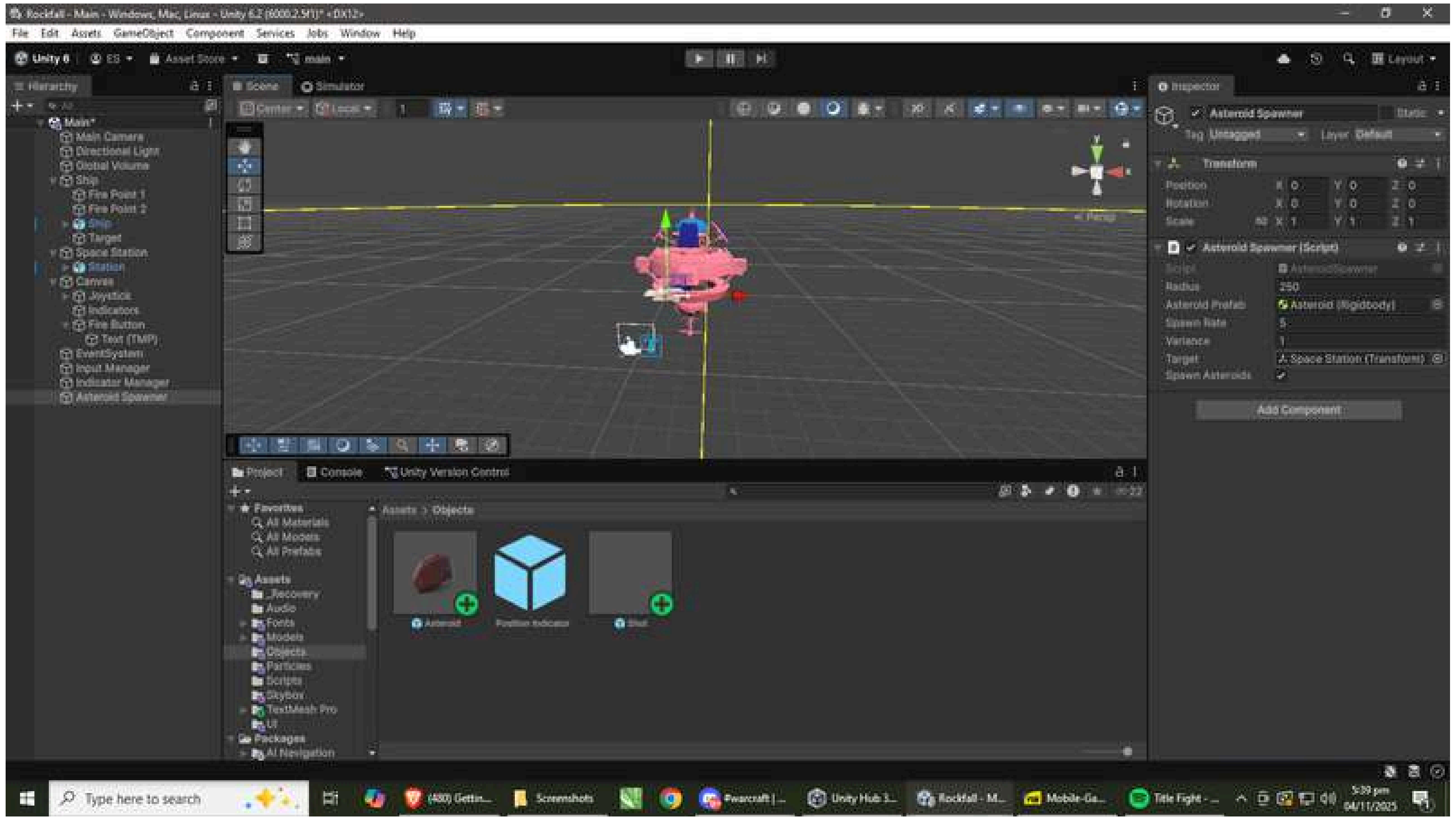
Asteroid Spawner
Tag Untagged Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 0.1 Z 1

Asteroid Spawner (Script)
Script AsteroidSpawner
Radius 250
Asteroid Prefab None (Rigidbody)
Spawn Rate 5
Variance 1
Target None (Transform)
Spawn Asteroids

STEP 56

Drag the Asteroid prefab that you just created into the Asteroid Prefab slot, and drag the Space Station object into the Target slot. Turn Spawn Asteroids on.



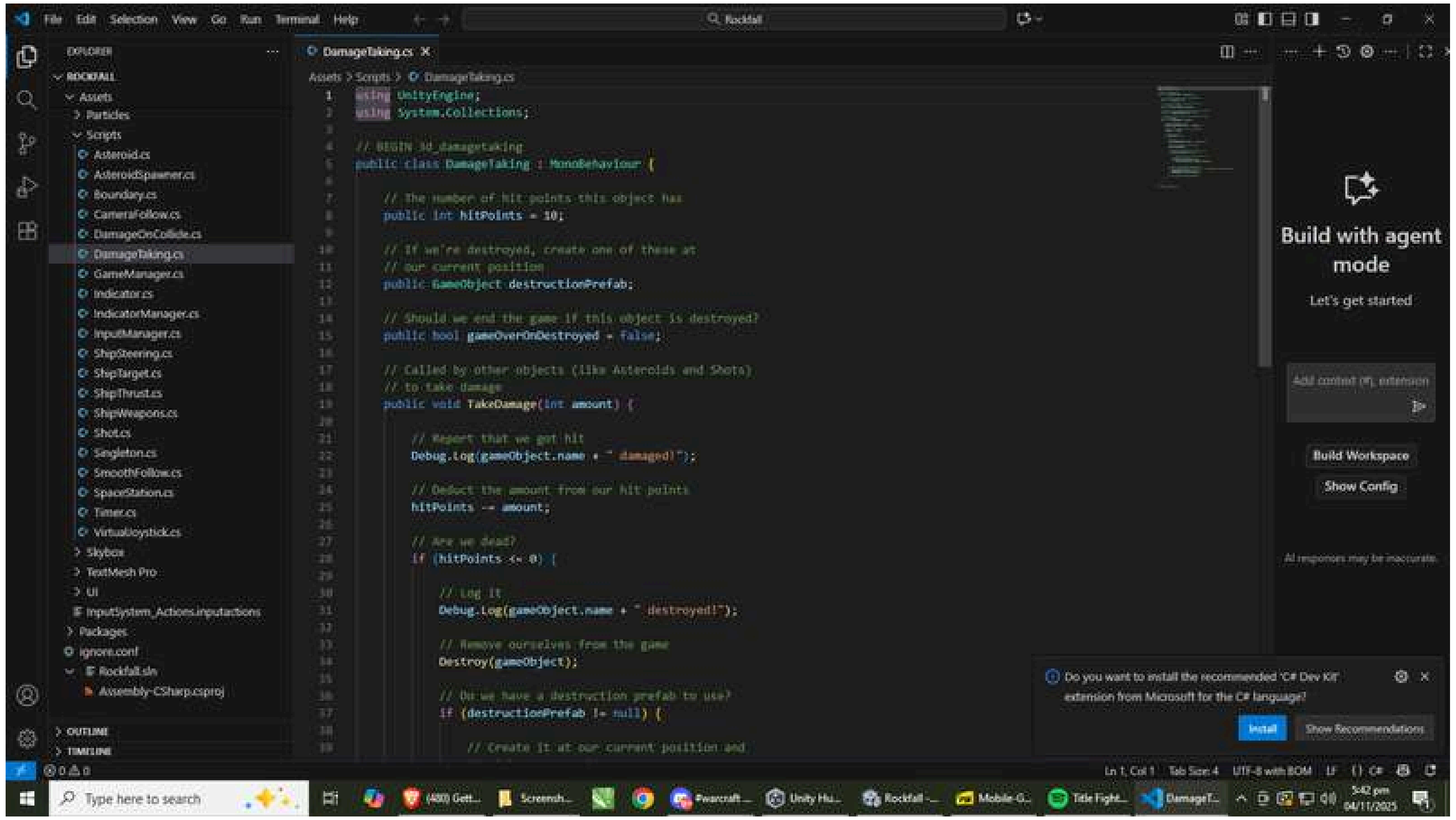
Asteroid Spawner
Tag Untagged Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Asteroid Spawner (Script)
Script AsteroidSpawner
Radius 250
Asteroid Prefab Asteroid (Rigidbody)
Spawn Rate 5
Variance 1
Target Space Station (Transform)
Spawn Asteroids ✓

STEP 57

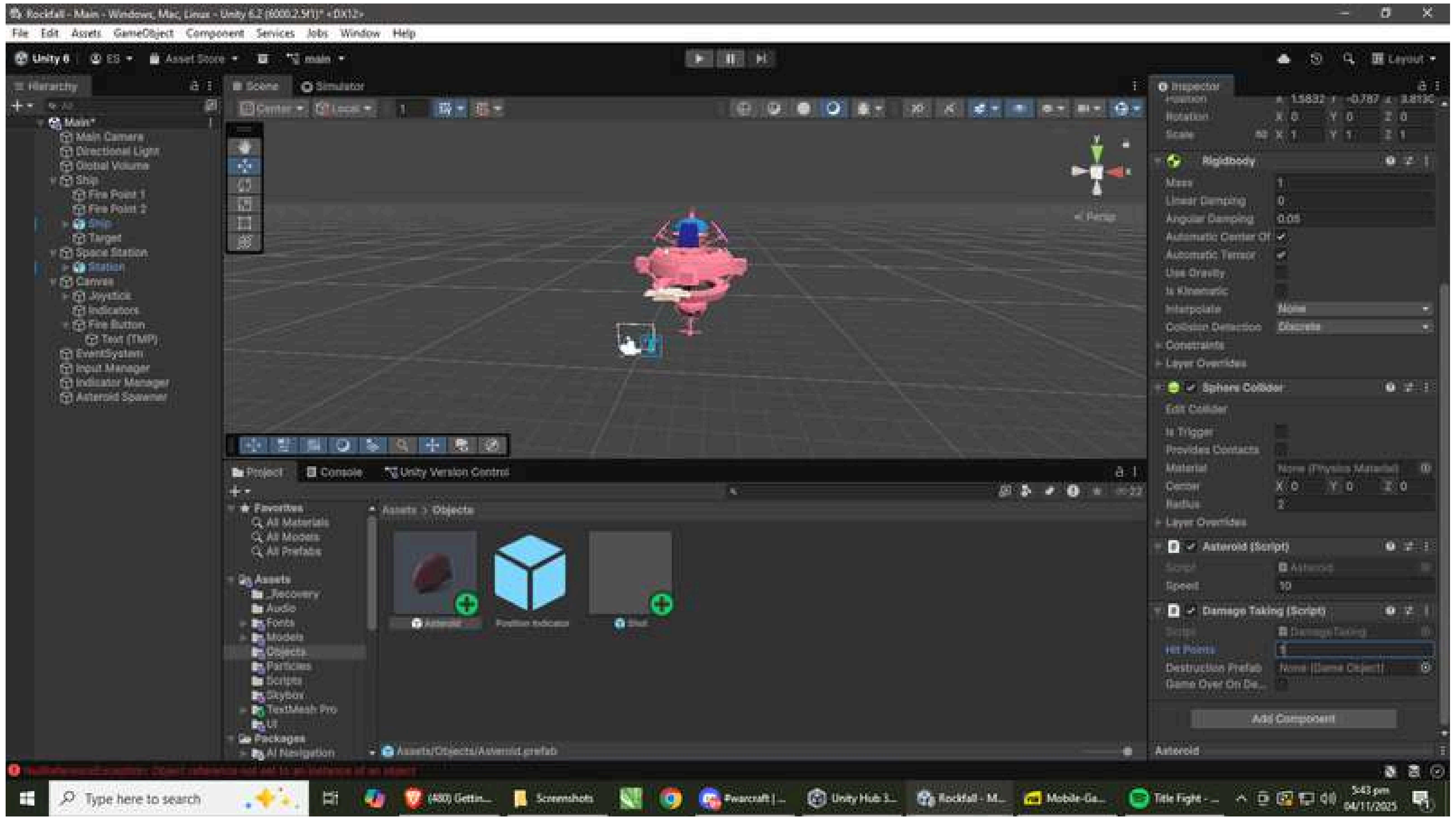
Select the Asteroid prefab in the Project pane, add a new C# script called Damage-Taking.cs to it, and add the following code to the file



```
using UnityEngine;
using System.Collections;

// BEGIN 3d_damagetaking
public class DamageTaking : MonoBehaviour {

    // The number of hit points this object has
    public int hitPoints = 10;

    // If we're destroyed, create one of these at
    // our current position
    public GameObject destructionPrefab;

    // Should we end the game if this object is destroyed?
    public bool gameOverOnDestroyed = false;

    // Called by other objects (like Asteroids and Shots)
    // to take damage
    public void TakeDamage(int amount) {

        // Report that we got hit
        Debug.Log(gameObject.name + " damaged!");

        // Deduct the amount from our hit points
        hitPoints -= amount;

        // Are we dead?
        if (hitPoints <= 0) {

            // Log it
            Debug.Log(gameObject.name + " destroyed!");

            // Remove ourselves from the game
            Destroy(gameObject);

            // Do we have a destruction prefab to use?
            if (destructionPrefab != null) {

                // Create it at our current position and
```

STEP 58

Change the asteroid’s Hit Points variable to 1. This will make the asteroid very easy to destroy



Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Rigidbody
Mass 1
Linear Damping 0
Angular Damping 0.05
Automatic Center Of ✓
Automatic Tensor ✓
Use Gravity
Is Kinematic
Interpolate None
Collision Detection Discrete
Constraints
Layer Overrides

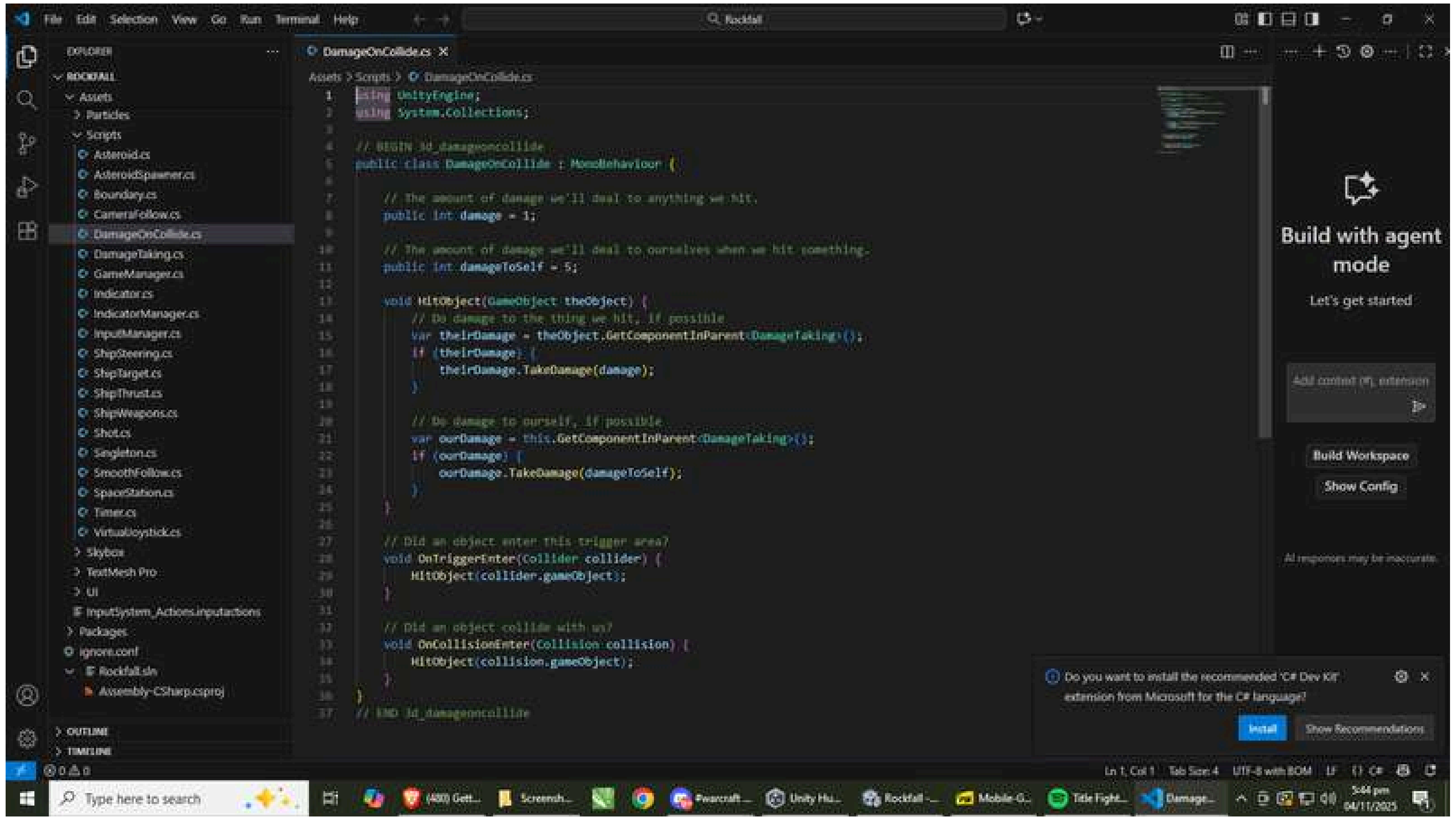
Sphere Collider
Edit Collider
Is Trigger
Provides Contacts
Material None (Physics Material)
Center X 0 Y 0 Z 0
Radius 2
Layer Overrides

Asteroid (Script)
Script Asteroid
Speed 10

Damage Taking (Script)
Script DamageTaking
Hit Points 1
Destruction Prefab None (Game Object)
Game Over On De...

STEP 59

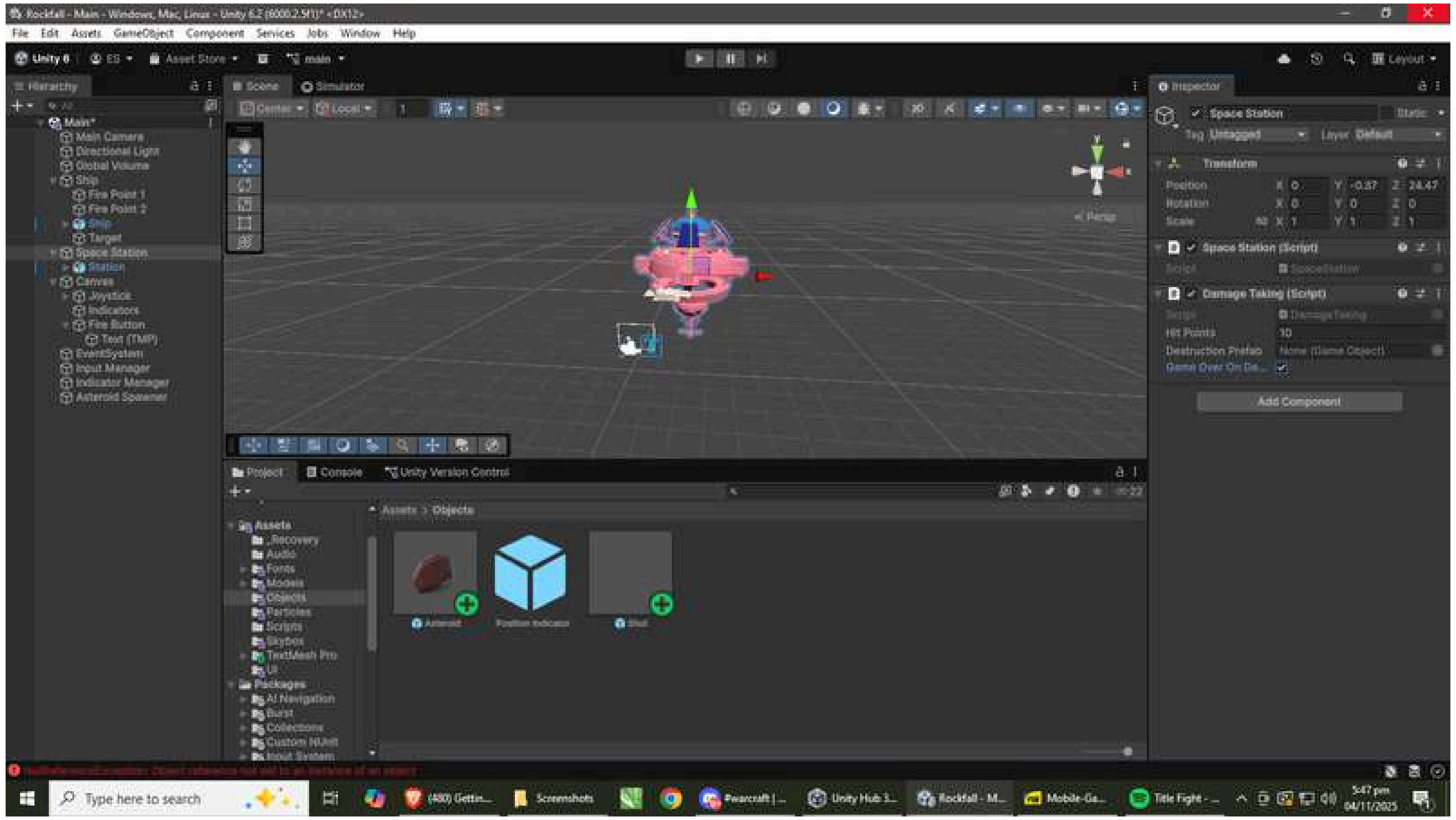
Select the Shot prefab, add a new C# script called DamageOnCollide.cs to it, and add the following code to the file.



```csharp
using UnityEngine;
using System.Collections;

// BEGIN 3d_damageoncollide
public class DamageOnCollide : MonoBehaviour {

    // The amount of damage we'll deal to anything we hit.
    public int damage = 1;

    // The amount of damage we'll deal to ourselves when we hit something.
    public int damageToSelf = 5;

    void HitObject(GameObject theObject) {
        // Do damage to the thing we hit, if possible
        var theirDamage = theObject.GetComponentInParent<DamageTaking>();
        if (theirDamage) {
            theirDamage.TakeDamage(damage);
        }

        // Do damage to ourself, if possible
        var ourDamage = this.GetComponentInParent<DamageTaking>();
        if (ourDamage) {
            ourDamage.TakeDamage(damageToSelf);
        }
    }

    // Did an object enter this trigger area?
    void OnTriggerEnter(Collider collider) {
        HitObject(collider.gameObject);
    }

    // Did an object collide with us?
    void OnCollisionEnter(Collision collision) {
        HitObject(collision.gameObject);
    }
}
// END 3d_damageoncollide
```

STEP 60

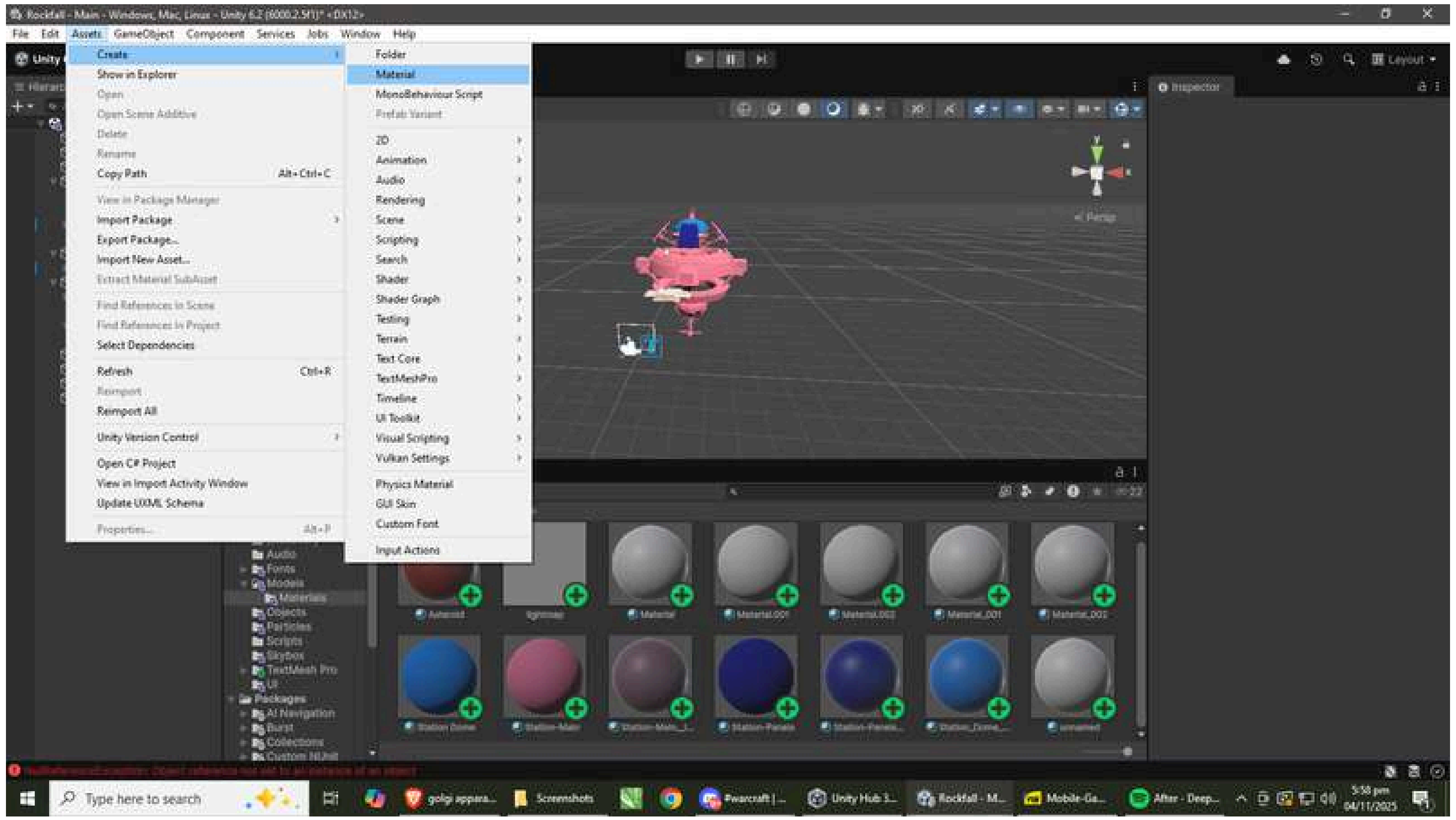
Select the Space Station, and add a DamageTaking script component. Turn on Game Over On Destruction.


Rockfall - Main - Windows, Mac, Linux - Unity 6.2 (6000.2.5f1)* <DX12>

Inspector: Space Station
Tag Untagged Layer Default
Transform
Position X 0 Y -0.37 Z 24.47
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Space Station (Script)
Script SpaceStation
Damage Taking (Script)
Script DamageTaking
Hit Points 10
Destruction Prefab None (Game Object)
Game Over On De... ✓

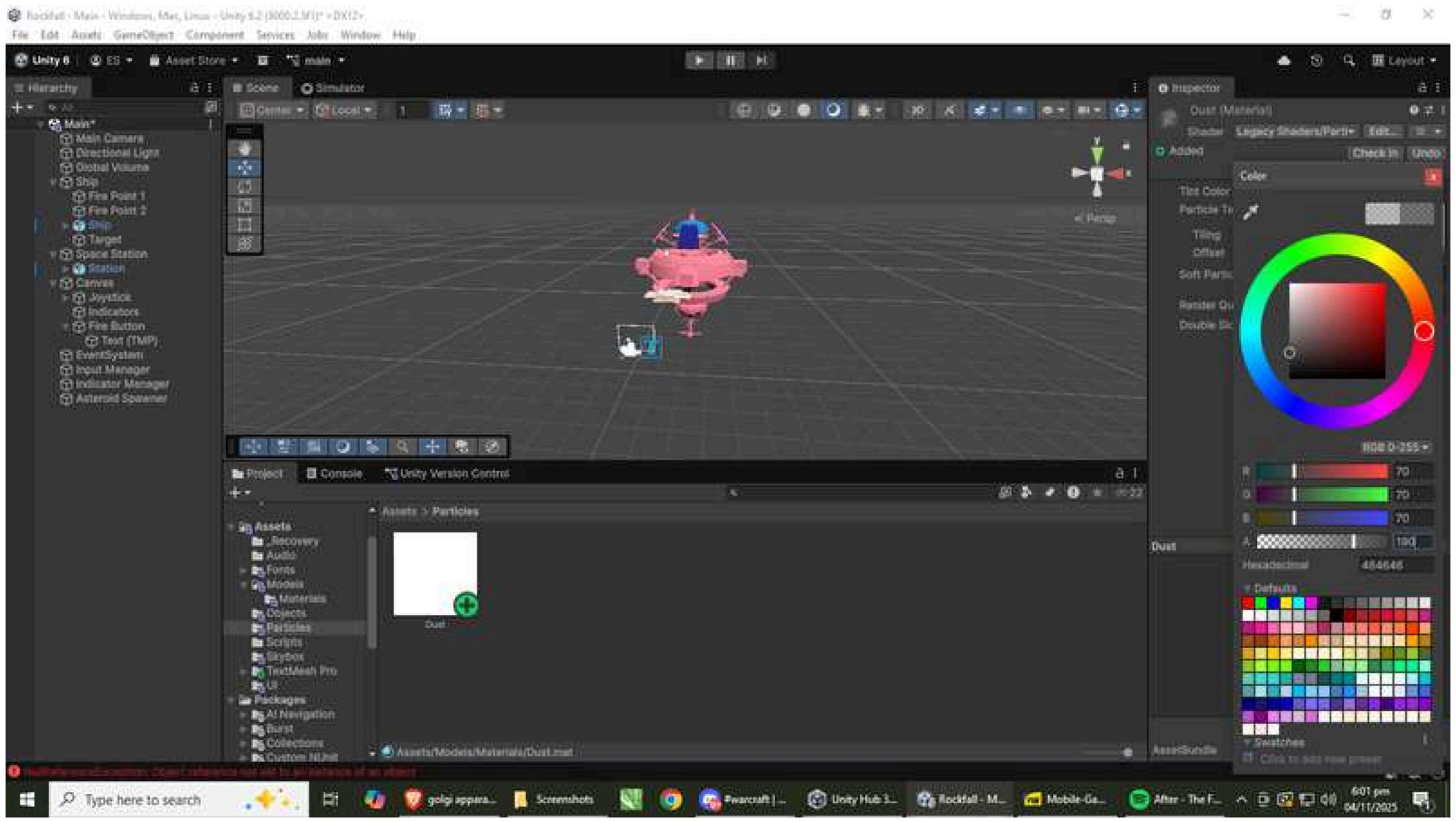
STEP 61

Open the Asset menu, and choose Create → Material. Name the new material “Dust”.



STEP 62

Select the material and change its shader to Particles/Additive. Next, drag the Dust texture into the Particle Texture slot. Set the tint color to a semiopqaue dark gray by clicking on the Tint Color slot and selecting a color. If you'd prefer to enter specific values, enter these: (70, 70, 70, 190)



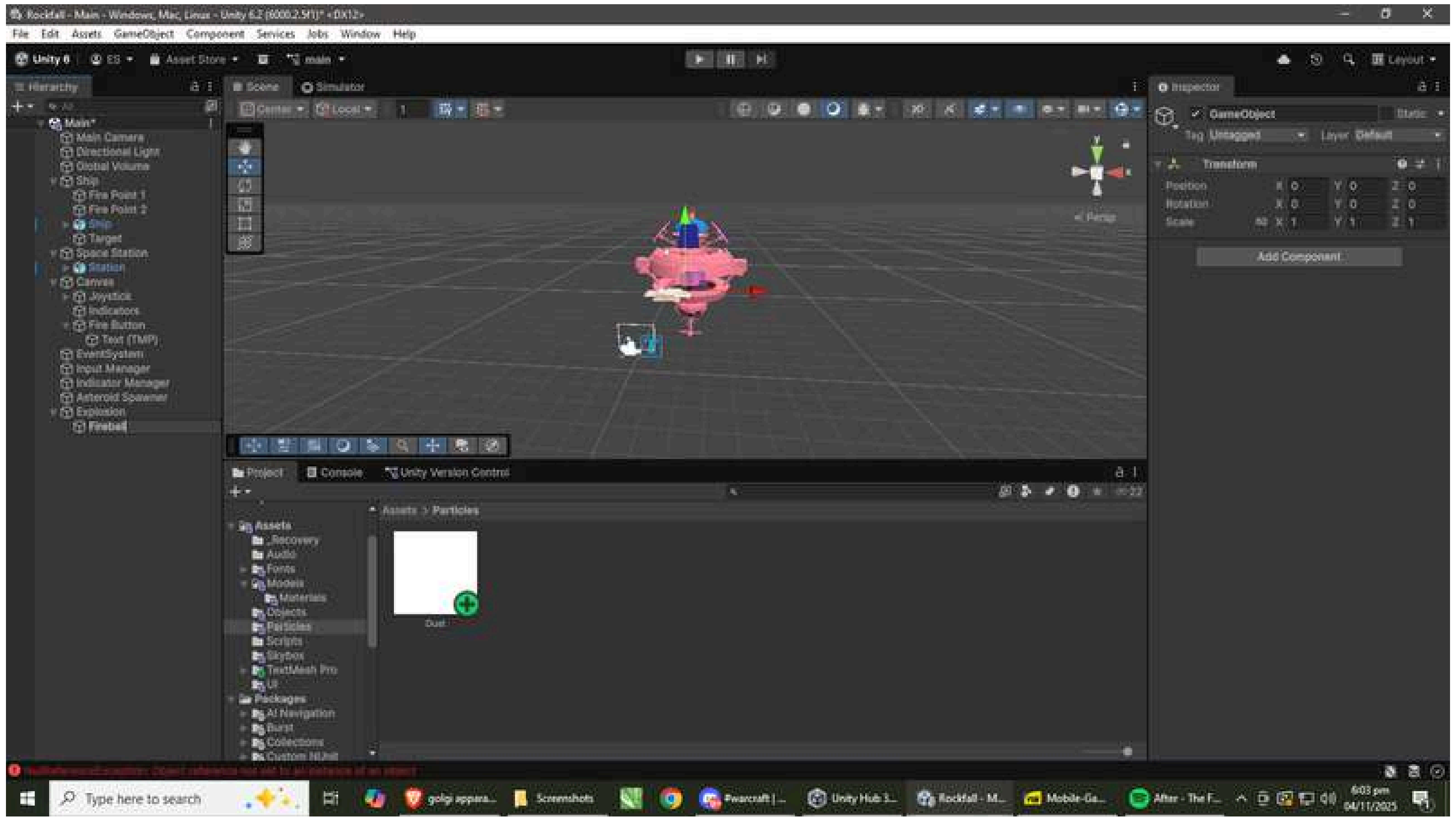
Shader: Legacy Shaders/Parti

R 70
G 70
B 70
A 190
Hexadecimal 464646

Assets/Models/Materials/Dust.mat

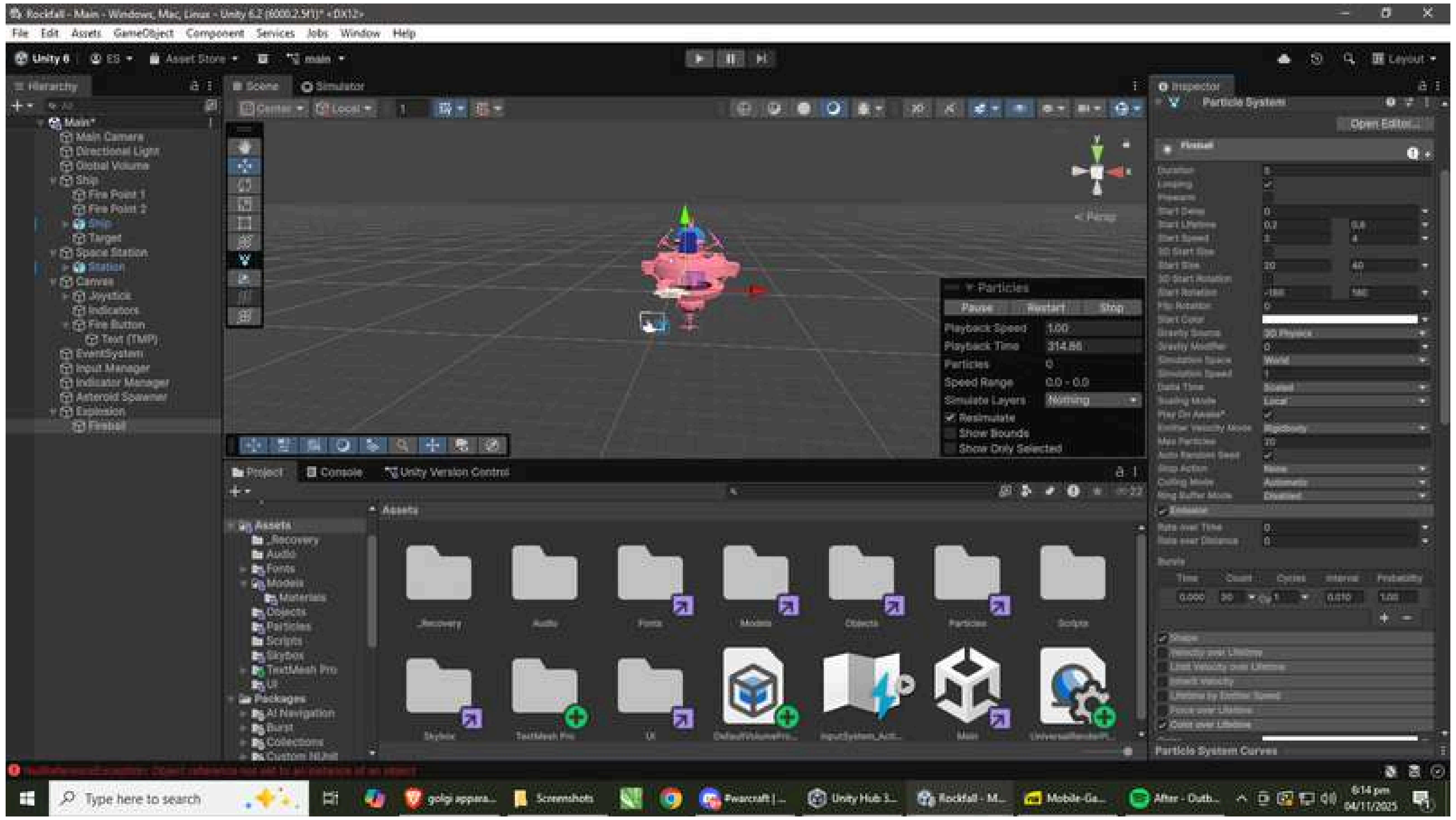
STEP 63

Create a new empty object, and name it “Explosion”. Create a second empty object, and name it “Fireball”. Make this object a child of the Explosion object.



STEP 64

Select the Fireball, and add a new Particle System component.



Particles: Pause, Restart, Stop

Playback Speed	1.00
Playback Time	314.86
Particles	0
Speed Range	0.0 - 0.0
Simulate Layers	Nothing
Resimulate	✓
Show Bounds	
Show Only Selected	

Duration	5	
Looping	✓	
Prewarm		
Start Delay	0	
Start Lifetime	0.2	0.6
Start Speed	3	4
3D Start Size		
Start Size	20	40
3D Start Rotation		
Start Rotation	-180	180
Flip Rotation	0	
Start Color		
Gravity Source	3D Physics	
Gravity Modifier	0	
Simulation Space	World	
Simulation Speed	1	
Delta Time	Scaled	
Scaling Mode	Local	
Play On Awake*	✓	
Emitter Velocity Mode	Rigidbody	
Max Particles	20	
Auto Random Seed	✓	
Stop Action	None	
Culling Mode	Automatic	
Ring Buffer Mode	Disabled	

Emission

Rate over Time	0
Rate over Distance	0

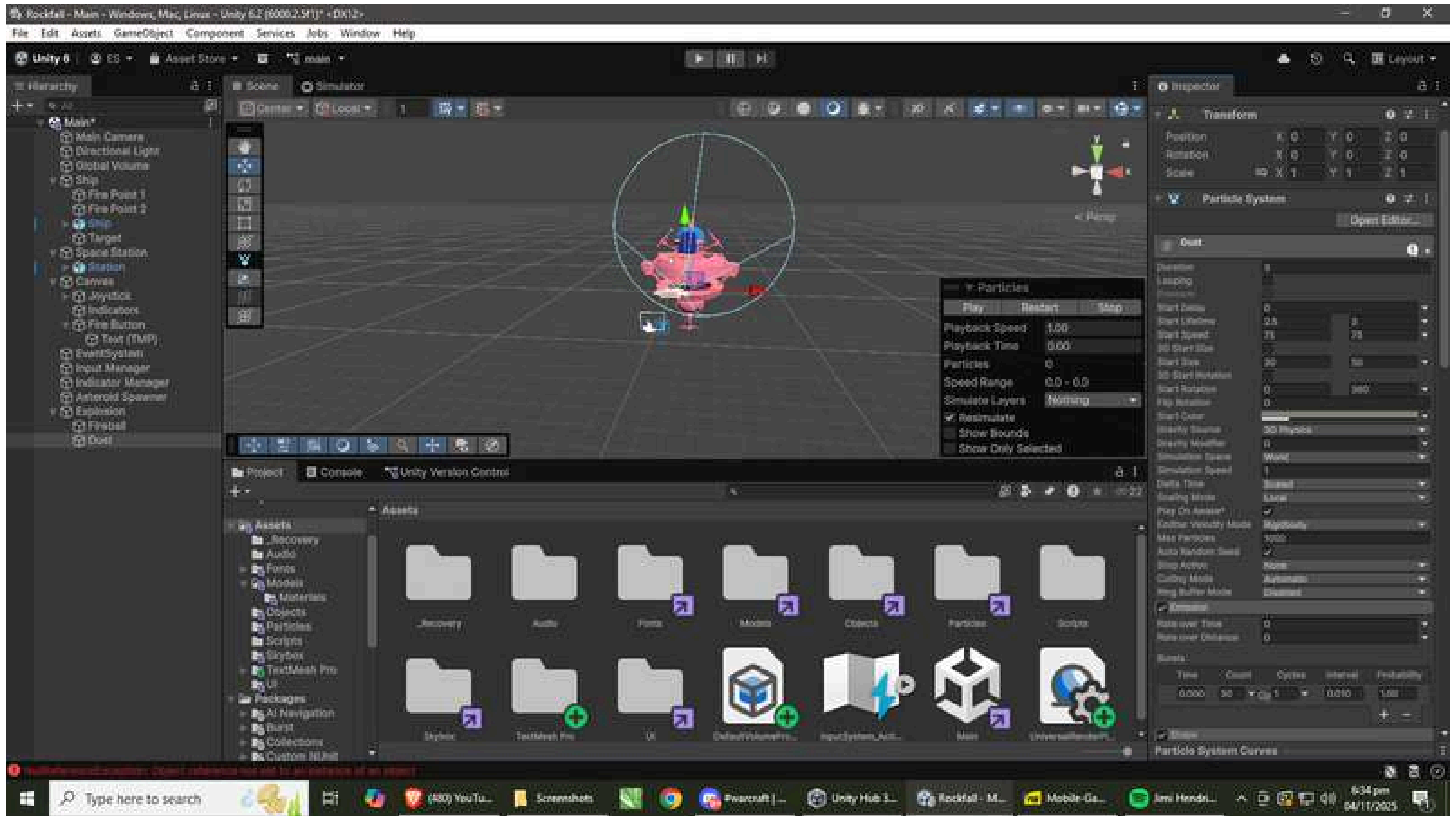
Bursts

Time	Count	Cycles	Interval	Probability
0.000	30	1	0.010	1.00

✓ Shape
Velocity over Lifetime
Limit Velocity over Lifetime
Inherit Velocity
Lifetime by Emitter Speed
Force over Lifetime
✓ Color over Lifetime

STEP 65

Make an empty game object, and name it “Dust”. Make it a child of the Explosion object. Add a new Particle System component, and set it up.



Particle System — Dust

Duration	5	
Start Delay	0	
Start Lifetime	2.5	3
Start Speed	75	75
Start Size	30	50
Start Rotation	0	360
Flip Rotation	0	
Gravity Source	3D Physics	
Gravity Modifier	0	
Simulation Space	World	
Simulation Speed	1	
Delta Time	Scaled	
Scaling Mode	Local	
Play On Awake*	✓	
Emitter Velocity Mode	Rigidbody	
Max Particles	1000	
Auto Random Seed	✓	
Stop Action	None	
Culling Mode	Automatic	
Ring Buffer Mode	Disabled	

Emission

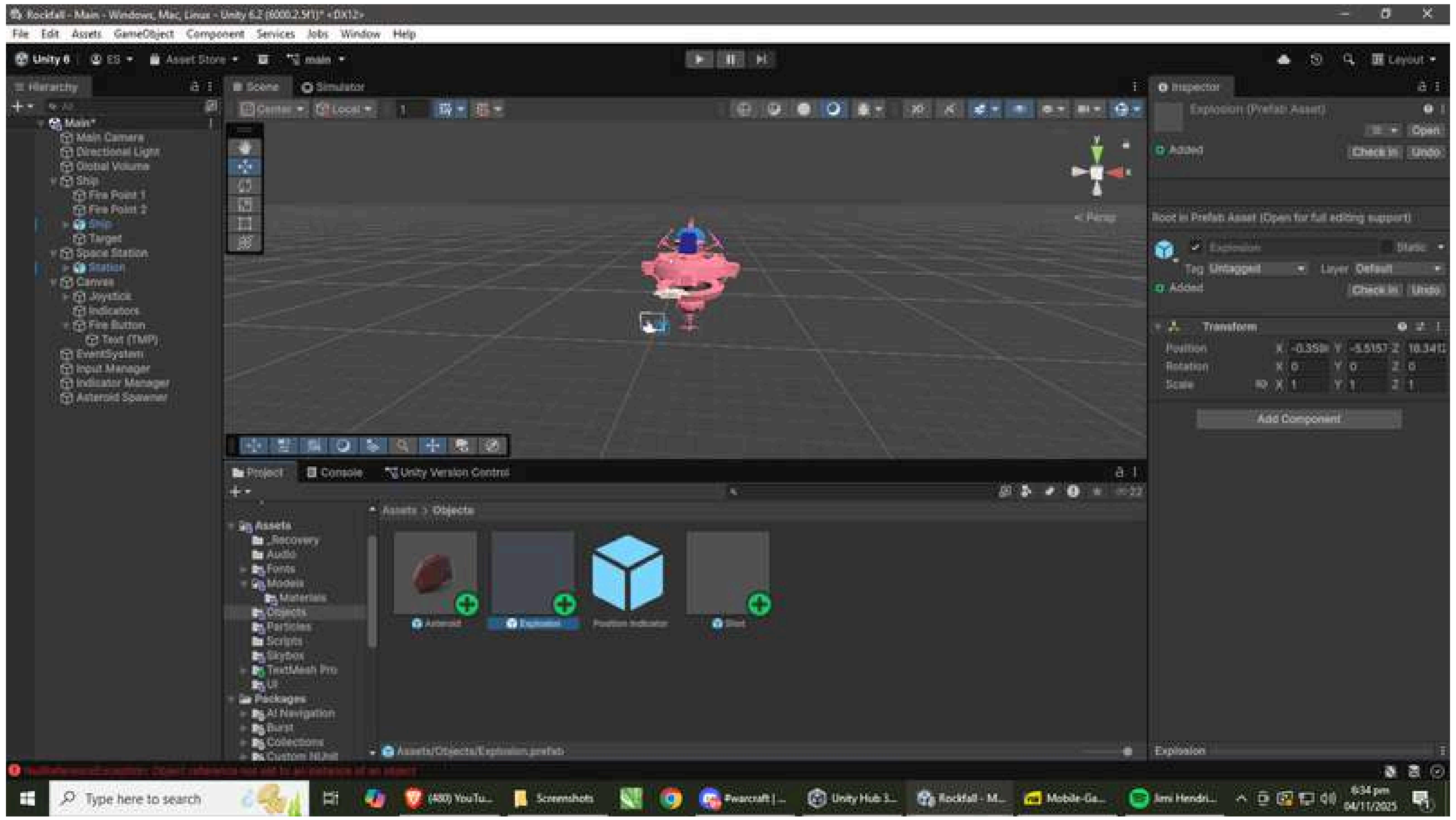
Rate over Time	0
Rate over Distance	0

Bursts

Time	Count	Cycles	Interval	Probability
0.000	30	1	0.010	1.00

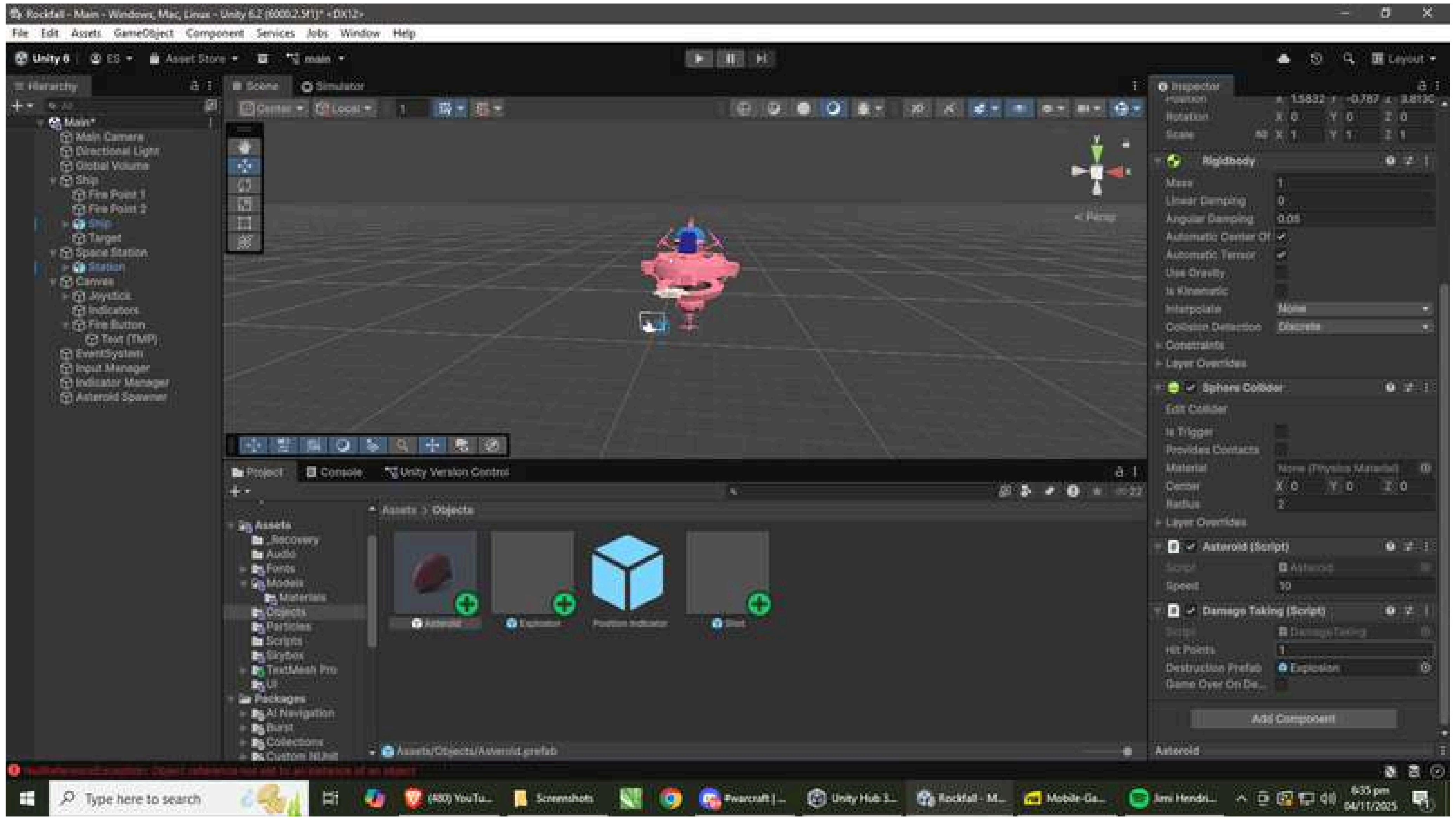
STEP 66

Drag the Explosion object into the Project pane, and then remove it from the scene.



STEP 67

Select the Asteroid prefab, and drag the Explosion into the Destruction Prefab slot.



Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Rigidbody
Mass 1
Linear Damping 0
Angular Damping 0.05
Automatic Center Of ✓
Automatic Tensor ✓
Use Gravity
Is Kinematic
Interpolate None
Collision Detection Discrete

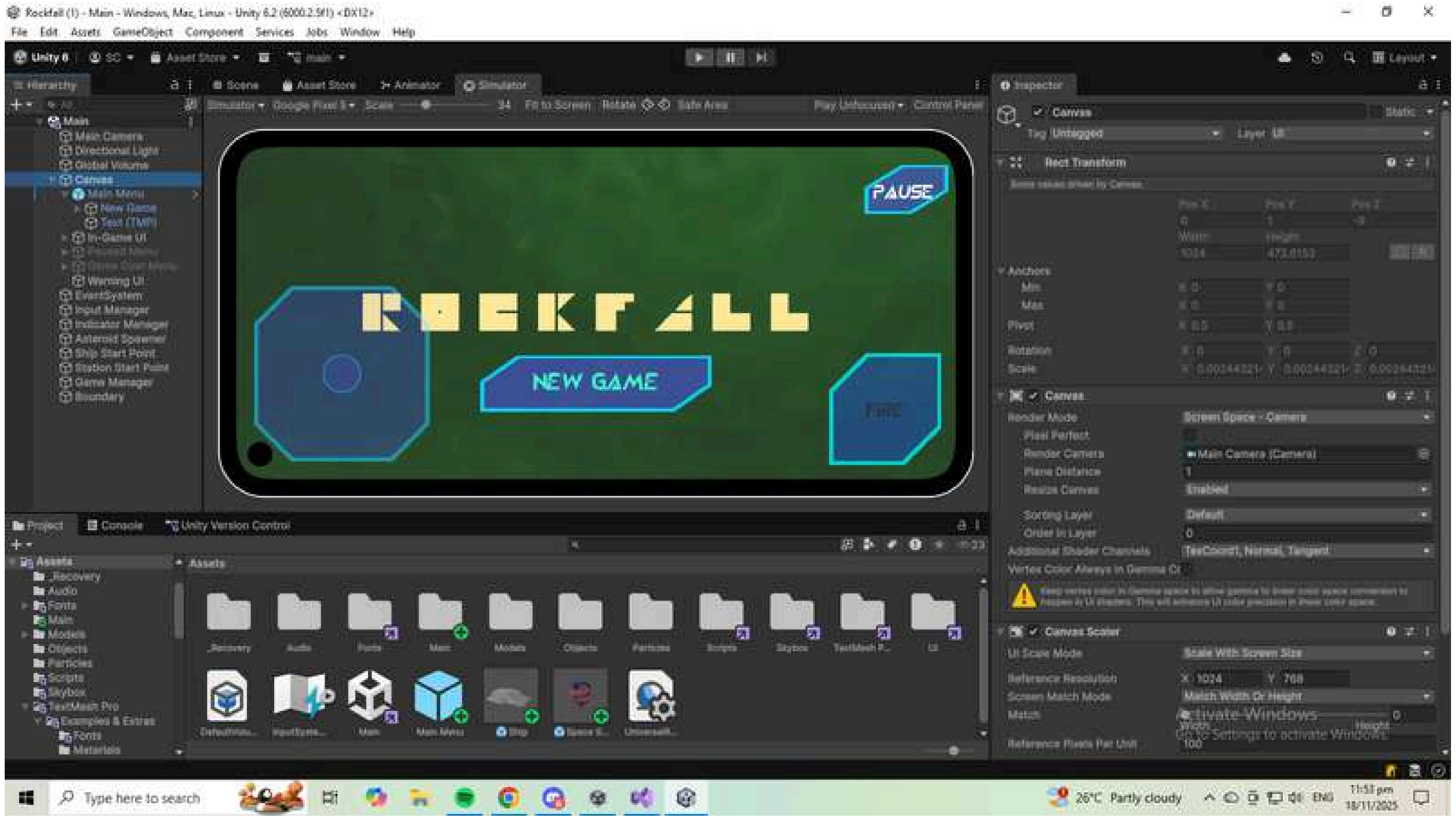
Sphere Collider
Is Trigger
Provides Contacts
Material None (Physics Material)
Center X 0 Y 0 Z 0
Radius 2

Asteroid (Script)
Script Asteroid
Speed 10

Damage Taking (Script)
Script DamageTaking
Hit Points 1
Destruction Prefab Explosion
Game Over On De...

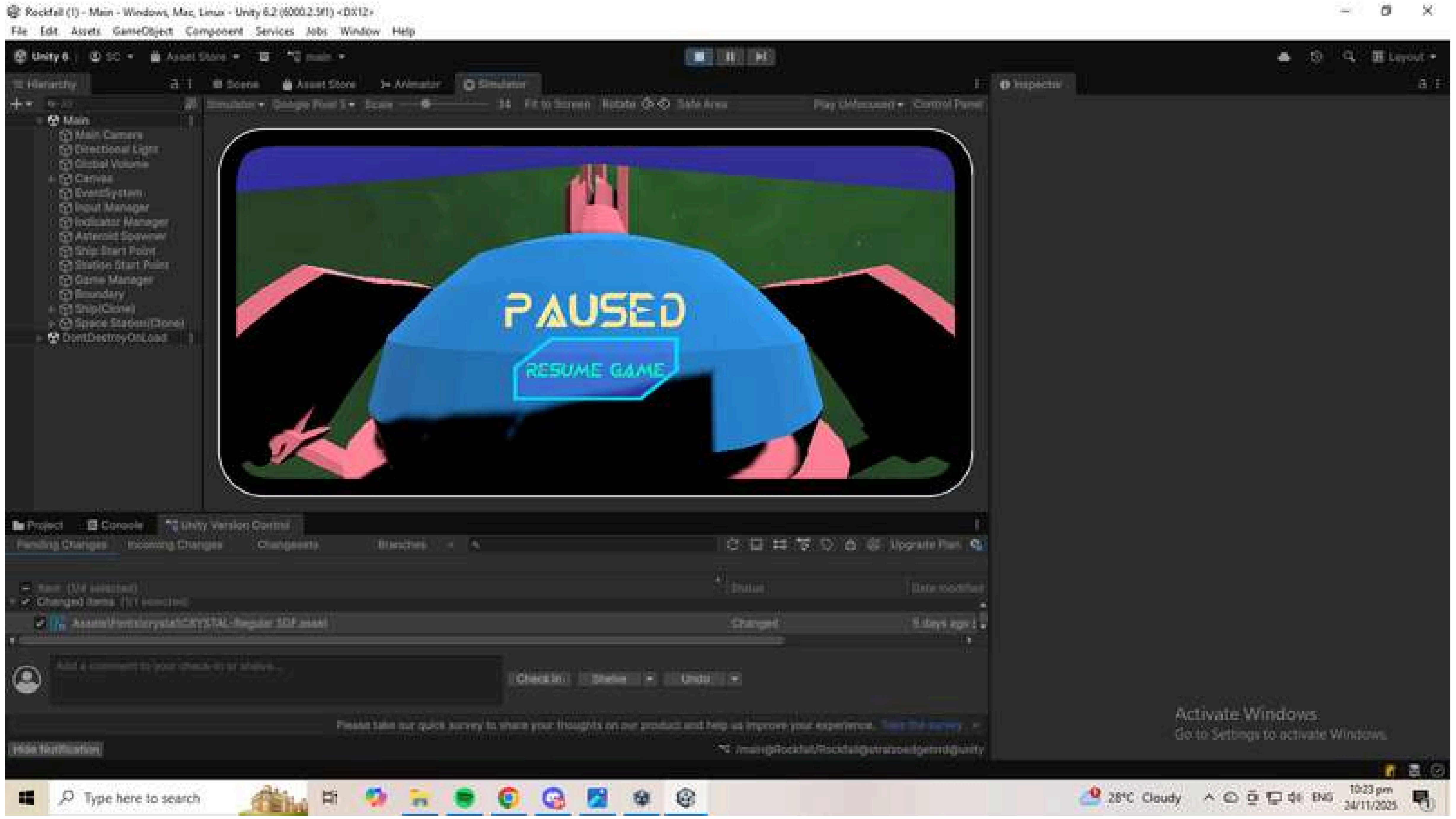
STEP 68

Add a Main Menu before the game starts



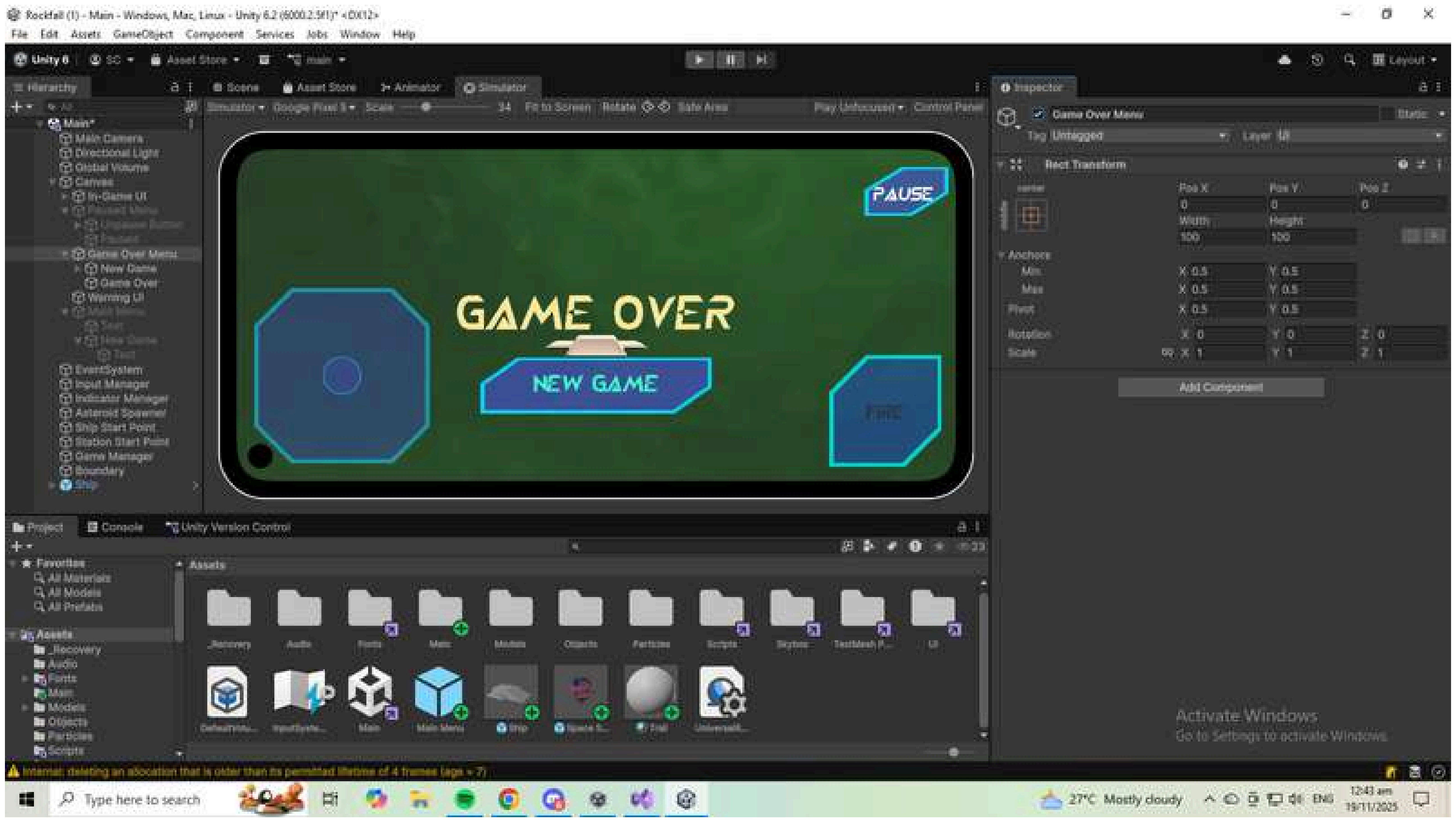
STEP 69

Add a Pause menu



STEP 70

Add a Game Over menu



Game Over Menu — Tag: Untagged — Layer: UI

Pos X 0 — Pos Y 0 — Pos Z 0
Width 100 — Height 100
Anchors Min X 0.5 Y 0.5
Max X 0.5 Y 0.5
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

```
Internal: deleting an allocation that is older than its permitted lifetime of 4 frames (age = 7)
```

THANK YOU FOR PLAYING